

# **Novel Surface Rendering and Object Registration Methods for Three Dimensional Medical Imaging:**

The “SpiderWeb” Surface Algorithm and the “Pointers”  
Technique for Integrating Multimodal Images.

**Daniel B. Karron**

A dissertation in the Department of Applied Science, submitted to the  
faculty of the Graduate School of Arts and Science in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy, at New York University.

**December 1992**

Approved : \_\_\_\_\_  
Professor Samuel J. Williamson  
Research Advisor

(This page intentionally left blank.)



**This is a copy of the dissertation as filed with the New  
York University Recorder on**

**January 8, 1993.**

---

This is version **0.04** , generated February 26, 1993.

Please address all correspondence to (**me !**) :

**Dan Karron, Ph.D.**

**Lab :**

New York University Medical Center

560 First Avenue

New York, New York, 10016

*e-mail* : karron@nyu.edu

*office phone* : 212 263 5210

**Home :**

333 East 30th Street, Apartment 21H

New York, New York, 10016

*home phone* : 212 779 0848

This dissertation will be published spring 1993 by University Microfilms.

Various minor corrections after the filing have been made. Each correction increments the version number.

This is a version using odd/even margins for double sided printing to conserve paper.

**Copyright ©1992 by Daniel B. Karron.**

**All Rights Reserved.**

**Patents Applied For by New York University and  
Daniel B. Karron.**



# Abstract

Daniel B. Karron.

New York University, Department of Applied Science.

Research Adviser : Professor Samuel J. Williamson.

**Keywords:** Three Dimensional Computer Graphics, Medical Imaging, Magnetic Resonance Imaging, Magnetoencephalography, Magnetic Source Imaging, Surface Rendering, Isosurface rendering, Image Registration, Object Registration, SpiderWeb Algorithm, Pointers Technique, CoordAlign Software Program, Blobulator Technique.

The goal of this dissertation is to develop mathematical and computer graphics techniques to enable precise and accurate combination of data from different medical imaging modalities into a single coordinate system for analysis and display.

Specifically, the problem addressed was to develop, test, and use techniques to combine magnetoencephalographic magnetic source imaging (MSI) data with magnetic resonance imaging (MRI) stacks of slices.

The “Pointers” technique was developed for interpolating the precise location of fiducial landmarks between slices in stacks of MRI slices. The “Pointers” are devices that use a material to produce a strongly contrasting converging pattern of dots and lines in the slices. This is used to precisely interpolate cardinal landmarks coordinate system. Various software and mathematical techniques, including the “Blobulator”, were developed to precisely locate a Pointers traces in two dimensional slices. These locations on slices were used to construct optimal lines that join each sequence of blobby mark in three dimensional space between the slices to ideally intersect at a cardinal landmark. A best estimate for the precise location of a cardinal landmark in space, independent of the slice orientation, is solved from the system of lines.

In addition, the “SpiderWeb Algorithm”, a new and topologically correct method of constructing three dimensional isosurfaces from volume density and volume gradient data, was developed. It is used, among other things, to reconstruct the skin isosurface for visually determining cardinal or fiducial landmarks. The advantage

of this approach is that no Pointers hardware need be worn by the subject when recording a MRI study. In addition, brain surfaces, and other anatomic boundaries can be determined for further physical modeling studies.

These techniques have been applied to locate the neuronal sources of occipital alpha rhythm in human subjects. Data that define the positions of sources of individual spindles of alpha rhythm are related to features of cortical anatomy provided by MR Images. An analysis shows that they cluster near individual sulci of cerebral cortex by lying within the visual area of the longitudinal fissure.



# Acknowledgements

I would like to thank my adviser, Professor **Sam Williamson** for his patience, inspiration, and enthusiasm. It has been an honor to study under, and work with him for these past few years. I would like to thank Professor **Gabe Miller** for his encouragement, help, and financial support in seeing me through the Applied Science Ph.D. program. Thanks to Professor **Marty Hoffert** for his leadership during my years with the Applied Science Department, and for showing me that all *real* problems can be reduced to a system of ODE's or PDE's. Thanks to **Court Cutting**, M.D., for showing me *real* biomedical engineering problems, and that each is basically simple if you think about it for a year or two. Also, for the continued use of his NYU laboratory space at all hours of the night. Professor **Bud Mishra**, for finding my questions interesting in his robotics class, and for reaching out to me, and being a true friend and colleague. Also, for his help and interest when I would bring research problems to him seeking to tap into his vast theoretical insight. Also for hot gossip. Thanks to **Ross Smith**, Ph.D., M.D., for keeping the NYU Medical Center network healthy, wired, running, and fast. Also, for the old days when we were sneaking wires around the campus (and not getting caught). To Professor **Ted Coons**, Ph.D., who took an interest in a confused, skinny kid in his Physiological Psychology class so *many* years ago. To Dr. **Greg Baumann**, Ph.D., for believing in me and supplying emergency funding when no one else would/could/ or had. To **Gene Grossi**, M.D., for dropping in between doing his heart cases and taking such a strong friendly interest in my research problems. Also, for dragging me along to go sailing on his boat, the **Marginal**, on New York harbor. To Professor **Jim Cox**, for thinking I was smarter than I am. And also for hanging out with me all night at the Medical Center working on the proofs that appear in Section 4. To **Charlie Hennoch**, Ph.D., **Ed Weinberger**, Ph.D., **Zhong-Lin Lü**, Ph.D., **Murlai Kadiramangalam**, Ph.D. (Who *told* me not to take the French exam), and **Alan Kalvin**, Ph.D. for showing me that it was possible to get a Ph.D. at NYU. To **Seth Potter** Ph.D., for being a true classmate and real friend through the classwork, written exam, oral exam. Thanks to **David Dean**, Ph.D., for being my friend and the laboratory communicator. Without him I would have cut off my nose to spite my face more than once. I would like to thank **Bill Lorensen** for his inspiration and the thought that, perhaps, I might have something to contribute to an area of computer graphics that he and Harvey Cline virtually invented. Also I would like to thank **Bruce (Bear) Giles** for critical and thoughtful commentary by e-mail in middle of the night. I would also like to thank my friends and colleagues at Silicon Graphics for their interest and support, including **Allan Trimble**, **Jim Helman**, Ph.D., and **John Campbell**, Ph.D. I also thank the **Second Avenue Deli** for the realization of the “2–4 Voxel face hit rule” . I would

like to thank **Tyler Volk**, Ph.D., for his cheery encouragement, and for setting a role model. I would like to thank **Sherryl Stewart**, Ph.D., whom as yet, I have never met, for stimulating e-mail in the middle of the night. Also, I would like to thank **Jeff Stevenson** for his expert and professional help with the **Philips GYROSCAN** Magnetic Resonance Imaging system. I acknowledge the cooperation of **New York University Department of Radiology** for letting Jeff help me and for time on the GYROSCAN. To **Cleve Backster**, for getting me started in High School. He gave me the idea that, perhaps, what most people considered common knowledge was completely wrong. I also owe thanks to **Joseph G. McCarthy** M.D., for *blasting* me out of computer systems administration and into basic science. I would like to thank **Ed Friedman** for his friendship over the many years I have been hanging around the Academic Computing Facility. I thank **Frank Lopresti** admitting me to his academy of computer *nudnicks* and their regular trips to **Chinatown**. To **Adam Stark** for scanning the facial photographs that appear in Section 6. To **John Simon** at **School of Visual Arts** for scanning some additional illustrations at the *very* last moment. I also would like to pay tribute to Professor **Max Goldstein** for his friendship and cooperation in getting the Medical Center on the network. May he Rest In Peace.

Additionally, I acknowledge partial support in the form of an equipment loan from **Silicon Graphics, Inc.** through Allan Trimble.

This research was supported in part by grants from **New York University Medical Center Department of Surgery** and the **Surgical Consultants Research Fund**.

This work was also supported in part from funds from the **Air Force Grant AFOSR-91-0401** to the Neuromagnetism Laboratory.

This work was also supported in part by funds from the **National Science Foundation** grants CDA-9018673 and CDA-9022527 and CCR-9002819 to Bud Mishra.

This dissertation was printed on an old Hewlett Packer Laser Jet 2 printer with a Xante Accel-a-Writer replacement motherboard. This replacement processor was provided by **Xante, Inc.** through its President, **Marty Kennedy**. I could never have printed the half-tone illustrations as clearly as they are rendered without the 600 × 600 dot per inch mode. This dissertation takes approximately 9 hours to print.



# Dedication

Thanks to my wife, **Gail**, for sharing me with Science and for her support of my returning to full time student-ship to complete my Ph.D. Also, for her help in cleaning up my muddy prose. To my **Mother**, for my artistic temperament in the hard sciences. To my daughter, **Sarah**, for letting me put her to bed before returning to the medical school to do some work. To my brother, **Abe** for tea and sympathy in the middle of the night.



# Contents

|           |   |           |
|-----------|---|-----------|
| <b>I</b>  | <b>Introduction to Dissertation</b>                                 | <b>1</b>  |
| <b>II</b> | <b>The SpiderWeb Isosurface Algorithm</b>                           | <b>5</b>  |
| <b>1</b>  | <b>Introduction to the SpiderWeb Algorithm</b>                      | <b>7</b>  |
| 1.1       | Why Another Algorithm ? . . . . .                                   | 8         |
| 1.1.1     | Novel Features of SpiderWeb Algorithm . . . . .                     | 9         |
| 1.1.2     | Example Medical Illustrations . . . . .                             | 12        |
| 1.1.3     | Physical Measurements Confirm Fidelity . . . . .                    | 22        |
| <b>2</b>  | <b>Critical Review of the Literature</b>                            | <b>23</b> |
| 2.1       | Other Surface Rendering Algorithms . . . . .                        | 24        |
| 2.1.1     | Surface Rendering by Contour Rings . . . . .                        | 24        |
| 2.1.2     | Volume Visualization Approaches . . . . .                           | 26        |
| 2.1.3     | Surface Rendering Algorithms . . . . .                              | 27        |
| 2.1.3.1   | Problems in Surface Algorithms Acknowledged in Literature . . . . . | 32        |
| <b>3</b>  | <b>The SpiderWeb Algorithm</b>                                      | <b>35</b> |
| 3.1       | Terminology and Definitions . . . . .                               | 35        |
| 3.1.1     | SpiderWeb For Two Dimensional Contour Line Algorithm . .            | 39        |
| 3.1.1.1   | Mathematical properties of functions and terminology                | 39        |
| 3.1.1.2   | Only one hit is possible along an edge . . . . .                    | 41        |

|          |  |    |
|----------|--|----|
| 3.1.1.3  | One hit on a face is not possible . . . . .                    | 43 |
| 3.1.1.4  | Two hits on a face yield unambiguous geometry . . .            | 45 |
| 3.1.1.5  | Three hits on a face is impossible . . . . .                   | 45 |
| 3.1.1.6  | Four Hits on a Boxel Face . . . . .                            | 48 |
| 3.1.1.7  | Connected Boxel Faces . . . . .                                | 50 |
| 3.1.1.8  | Connected Edges . . . . .                                      | 50 |
| 3.1.1.9  | SpiderWeb A Hedges . . . . .                                   | 50 |
| 3.1.1.10 | SpiderWeb B Chooses . . . . .                                  | 52 |
| 3.1.1.11 | The SpiderWeb B Choice . . . . .                               | 52 |
| 3.1.1.12 | Boxel edge-wise continuity . . . . .                           | 55 |
| 3.1.1.13 | Adaptive refinement . . . . .                                  | 55 |
| 3.1.1.14 | Triangular decomposition . . . . .                             | 58 |
| 3.1.1.15 | Other shape boxels . . . . .                                   | 66 |
| 3.1.2    | 2D SpiderWeb curve properties . . . . .                        | 66 |
| 3.1.2.1  | 2D Curvature Limitation with respect to Edge Length            | 66 |
| 3.1.2.2  | 2D Gradient strength . . . . .                                 | 69 |
| 3.1.2.3  | 2D Noise . . . . .   | 71 |
| 3.1.2.4  | Jordan Separation and Curve Theorem . . . . .                  | 71 |
| 3.1.3    | Uses of contour lines to preview 3D reconstruction . . . . .   | 73 |
| 3.1.3.1  | Summary of SpiderWeb A and B for 2D . . . . .                  | 74 |
| 3.2      | Spinning Three Dimensional Surfaces from Volume Data . . . . . | 75 |
| 3.2.1    | Three Dimensional Surfaces in Volume Data . . . . .            | 76 |
| 3.2.2    | Three Dimensional Voxel Definitions . . . . .                  | 77 |
| 3.2.2.1  | Properties of Hits . . . . .                                   | 81 |
| 3.2.2.2  | The Voxel Face-Wise Continuity Rule . . . . .                  | 85 |
| 3.2.2.3  | The Articulating Point . . . . .                               | 89 |
| 3.2.2.4  | Some properties of a surface tileing . . . . .                 | 90 |
| 3.2.2.5  | The Jordan Surface Theorem . . . . .                           | 90 |
| 3.2.2.6  | The Möbius Condition . . . . .                                 | 90 |

|         |  |     |
|---------|--|-----|
| 3.2.3   | Triangle Definitions . . . . .   | 93  |
| 3.2.3.1 | Spinning Triangles in a Voxel Illustrated . . . . .                                      | 98  |
| 3.2.3.2 | Comparing Volume and Surface Rendering Algorithms: Volume Occupancy and Depth Complexity | 114 |
| 3.2.3.3 | Simplifying SpiderWeb geometry . . . . .   | 118 |
| 3.2.3.4 | Fast Triangle Mesh Generation From SpiderWeb Tileing                                     | 125 |
| 3.2.4   | Properties of SpiderWeb Surfaces . . . . .   | 127 |
| 3.2.4.1 | 3D Noise and the Surface . . . . .   | 127 |
| 3.2.4.2 | 3D Gradient Strength of Surface . . . . .  | 127 |
| 3.2.4.3 | SpiderWeb Surfaces Satisfy Euler–Poincarè Law . . .                                      | 129 |
| 3.2.4.4 | Ellipsoid of minimum curvature . . . . .   | 131 |
| 3.2.5   | New dimensions in SpiderWeb topological analysis suggested .                             | 133 |
| 3.2.5.1 | Introduction . . . . .   | 133 |
| 3.2.5.2 | Isovalue Contour Lines on Isovalue Surfaces . . . . .                                    | 133 |
| 3.2.5.3 | Gradient Strength . . . . .  | 134 |
| 3.2.5.4 | Gradient Surface . . . . .   | 134 |
| 3.2.5.5 | Isogradient Surface projected on the Isodensity Surface                                  | 134 |
| 3.2.5.6 | Isocurvature surfaces projected on the isodensity surface                                | 134 |
| 3.2.5.7 | Extremal value paths through volume . . . . .  | 135 |
| 3.2.5.8 | Threshold value as an additional dimension . . . . .                                     | 135 |
| 3.2.5.9 | Segmentation and Geometry . . . . .  | 145 |
| 3.2.6   | Specific Claims for SpiderWeb Algorithm in 3 Dimensions . .                              | 147 |
| 3.2.7   | Overview of Algorithm : Macroscopic and Microscopic Properties                           | 150 |
| 3.2.7.1 | Overview : Microscopic . . . . .   | 150 |
| 3.2.7.2 | Overview : Macroscopic . . . . .   | 150 |

|            |  |            |
|------------|--|------------|
| <b>4</b>   | <b>Mathematical Proofs of Properties</b>   | <b>151</b> |
| 4.1        | Introduction and Strategy of Proofs . . . . .  | 151        |
| 4.2        | Proof of the Correctness of the SpiderWeb Algorithm . . . . .                              | 153        |
| 4.2.1      | Proof of SpiderWeb Correctness . . . . .   | 153        |
| 4.2.2      | Topological Preliminaries . . . . .  | 154        |
| 4.2.2.1    | Properties of Voxel Hits. . . . .  | 157        |
| 4.2.2.2    | Properties of simplicial complex (triangle mesh) . . .                                     | 158        |
| 4.2.2.3    | Boundedness by zero value assumption . . . . .   | 159        |
| 4.2.2.4    | Proof of SpiderWeb B . . . . .   | 163        |
| 4.3        | Concluding Remarks . . . . .   | 165        |
| <br>       |  |            |
| <b>III</b> | <b>The Pointers and Object Registration</b>  | <b>169</b> |
| <br>       |  |            |
| <b>5</b>   | <b>Implementation of the CoordAligner and the Pointers System in Hardware and Software</b> | <b>171</b> |
| 5.1        | Introduction to the “CoordAligner” and the “Pointers” . . . . .                            | 173        |
| 5.2        | Background and State of the Art of Magnetic Source Imaging . . . .                         | 174        |
| 5.2.1      | Brief Overview of SQUID Technology . . . . .   | 177        |
| 5.2.2      | Locating a Current Dipole . . . . .  | 179        |
| 5.2.2.1    | Volume Electrical and Magnetic Fields: Law of Biot and Savart . . . . .                    | 179        |
| 5.2.2.2    | Cellular Currents . . . . .  | 180        |
| 5.3        | Magnetic Resonance Imaging Physics and Technology . . . . .                                | 182        |
| 5.3.0.3    | MRI Pulse Sequence Overview . . . . .  | 185        |
| 5.3.1      | The Problem of Disparate Coordinate Systems in Medical Imaging . . . . .                   | 186        |
| 5.3.1.1    | Coordinate System Aliasing . . . . .   | 187        |
| 5.3.1.2    | Locating landmarks in MRI slices . . . . .   | 189        |
| 5.3.1.3    | Direct Estimation of Cardinal Landmarks on Slices .  | 189        |
| 5.3.1.4    | The Pointers for Cardinal Landmark Localization. .   | 190        |



|          |  |            |
|----------|--|------------|
| 5.3.1.5  | The SpiderWeb for Landmark Localization . . . . .                              | 194        |
| 5.3.1.6  | Combining Metric Systems: The “CoordAligner” . .                               | 194        |
| 5.3.1.7  | Experimental Validation of the Techniques . . . . .                            | 195        |
| 5.3.2    | Review of the State of the Art . . . . .                                       | 195        |
| 5.3.2.1  | The Direct Method for Finding Landmarks . . . . .                              | 198        |
| 5.3.2.2  | The Hat of Pelizzari . . . . .   | 198        |
| 5.3.2.3  | Qsh . . . . .  | 200        |
| 5.3.2.4  | Longitudinal Fissure as a Landmark . . . . .                                   | 202        |
| 5.3.3    | Other Registration systems . . . . .   | 204        |
| 5.4      | Cardinal Landmarks . . . . .   | 209        |
| 5.4.0.1  | The classical anatomic orientations: axial, coronal,<br>and sagittal . . . . . | 212        |
| 5.4.1    | The PPN Transform Algorithm . . . . .  | 212        |
| 5.4.2    | Computing the PPN Headframe Axes . . . . .                                     | 215        |
| 5.4.2.1  | Error Analysis of PPN Cardinal Landmarks . . . . .                             | 217        |
| 5.5      | The CoordAligner: A system to register stacks of image slices . . . .          | 224        |
| 5.5.1    | Pointers for Localization of Cardinal Landmarks . . . . .                      | 228        |
| 5.5.2    | Pointers Algorithm . . . . .   | 240        |
| 5.6      | Visual Localization of Cardinal Landmarks Provided by SpiderWeb .              | 246        |
| <b>6</b> | <b>Results: Accuracy of the Pointer and SpiderWeb Registration Method</b>      | <b>249</b> |
| 6.0.0.1  | Description of data sets . . . . .   | 251        |
| 6.0.1    | Results of Localization of the Cardinal Landmarks from the Skin                | 251        |
| 6.0.1.1  | Specifications for the Polhemus 3Space Digitizer Mea-<br>surements . . . . .   | 252        |
| 6.0.1.2  | Head Measurement Procedure for the Neuromagne-<br>tometer . . . . .            | 252        |
| 6.0.1.3  | Key to Polhemus digitizer probe measurement tables<br>headings . . . . .       | 253        |

|           |  |            |
|-----------|--|------------|
| 6.0.2     | Localization of Cardinal Landmarks in MRI Stacks . . . . .   | 256        |
| 6.0.2.1   | Results of Direct localization of Cardinal Landmarks<br>from MRI Slices . . . . .                              | 257        |
| 6.0.2.2   | Key to Direct slice measurements tables headings . .   | 257        |
| 6.0.2.3   | Key to Direct error distance table headings . . . . .  | 259        |
| 6.0.2.4   | Key to warpage table headings . . . . .  | 261        |
| 6.0.2.5   | Results of Pointer Localization of Cardinal Landmarks  | 263        |
| 6.0.2.6   | Results of Localization of Cardinal Landmarks on Spi-<br>derWeb Skin Isosurface . . . . .                      | 267        |
| <b>7</b>  | <b>Discussion</b>  | <b>283</b> |
| 7.0.2.7   | Summary of average measurement errors: Key to table  | 284        |
| 7.0.2.8   | Discussion of Summary Error Table . . . . .  | 286        |
| <b>IV</b> | <b>Neuromagnetic Brain Function Studies</b>  | <b>287</b> |
| <b>8</b>  | <b>Visualizing and Interpreting Neuromagnetic Data</b>   | <b>289</b> |
| 8.1       | Sphere Fitting of the Head . . . . .   | 289        |
| 8.1.1     | Introduction: Why fit a sphere inside the brain case ? . . . .   | 289        |
| 8.2       | The magnetic skullcap: a 3D Delaunay triangularization of MEG data<br>for visualization and modeling . . . . . | 292        |
| 8.3       | Forward Simulation Studies: Superficial Magnetic Fields Modeled from<br>Deep Electric Currents . . . . .       | 302        |
| 8.4       | Comprehensive Source Searching: An Inverse Simulation Study . . . .  | 304        |
| 8.4.1     | Introduction . . . . .   | 304        |
| 8.4.2     | Method . . . . .   | 306        |
| 8.4.3     | Parcelization of Image Surface . . . . .   | 306        |
| 8.4.3.1   | Comprehensive Source Searching Algorithm . . . . .   | 307        |
| 8.4.4     | Image Surface Parcelization . . . . .  | 309        |
| 8.4.5     | An Example of Applying CSS . . . . .   | 310        |



|           |  |            |
|-----------|--|------------|
| 8.5       | Discussion . . . . .   | 310        |
| <b>9</b>  | <b>Alphon Location by Registering MRI and MSI Coordinate Systems</b> | <b>313</b> |
| 9.1       | Introduction to Alphon Activity Localization . . . . .               | 313        |
| 9.2       | Introduction . . . . .   | 314        |
| 9.3       | Method . . . . .   | 316        |
| 9.4       | Results . . . . .  | 319        |
| 9.4.1     | Comparison with Narrow-Band Filtered Noise . . . . .                 | 319        |
| 9.4.2     | Location Stability . . . . .   | 319        |
| 9.4.2.1   | Discussion of Tables and Figures . . . . .                           | 320        |
| 9.5       | Discussion . . . . .   | 329        |
| <b>V</b>  | <b>Summary</b>   | <b>331</b> |
| <b>10</b> | <b>New algorithms developed</b>                                      | <b>333</b> |
| <b>11</b> | <b>Software written</b>  | <b>335</b> |
| 11.1      | Experiments conducted . . . . .                                      | 335        |
| <b>12</b> | <b>Concluding Discussion</b>   | <b>337</b> |
| <b>A</b>  | <b>Appendix</b>  | <b>339</b> |
| A.1       | PPN Headframe C Source Code . . . . .                                | 340        |
| A.2       | Best Point and Line C Source Code. . . . .                           | 342        |
| A.3       | Best Sphere C Source Code . . . . .                                  | 351        |
| A.4       | Sample Pointer Error Report For Analyzing Pointer Images . . . . .   | 355        |



# List of Figures

|      |   |    |
|------|---|----|
| 1.1  | An example of surface tileing errors in a prominent place in the literature.                    | 10 |
| 1.2  | Detail of surface tileing errors in preceding Figure. . . . .                                   | 11 |
| 1.3  | A representative sagittal MRI slice of the head. . . . .  | 14 |
| 1.4  | A sagittal MRI slice, shown as a <i>bas</i> relief. . . . .                                     | 15 |
| 1.5  | A stack of trimmed sagittal slices of the head. . . . .   | 16 |
| 1.6  | The skin isosurface of the head reconstructed from a stack of MRI slices.                       | 17 |
| 1.7  | The skin isosurface detail of the reconstructed nose and brow. . . . .                          | 18 |
| 1.8  | Skin isosurface detail of the ear. . . . .  | 19 |
| 1.9  | An artificial linear volume gradient with a few slices perturbed by random noise. . . . .       | 21 |
| 3.1  | A isocontour plot of an MRI axial slice of the brain to preview the SpiderWeb geometry. . . . . | 37 |
| 3.2  | A 4X magnification of previous MRI slice showing contour line detail                            | 38 |
| 3.3  | Linear interpolation of hit position on edge. . . . .   | 42 |
| 3.4  | Impossibility of 1 hit on a face. . . . .   | 44 |
| 3.5  | Impossibility of 3 hits on a face. . . . .  | 46 |
| 3.6  | Impossible surface configurations. . . . .  | 47 |
| 3.7  | Ambiguity in 2D Contour Lines. . . . .  | 49 |
| 3.8  | Faces are connected when they share an edge hit. . . . .  | 51 |
| 3.9  | Two topologies depending on knitting choices. . . . .   | 54 |
| 3.10 | Adaptive boxel sampling faces. . . . .  | 57 |

|      |  |     |
|------|--|-----|
| 3.11 | Adding an edge to decompose a square into triangles. . . . .   | 59  |
| 3.12 | Adding a new vertex can disambiguate the 4-hit face. . . . .   | 60  |
| 3.13 | Triangular boxels can only have 0 or 2 hits. . . . .   | 61  |
| 3.14 | Triangularization of a square boxel is not unique. . . . .   | 62  |
| 3.15 | One of two possible topologies that depend on the new vertex state. .  | 64  |
| 3.16 | The other two possible topologies that depend on the new vertex state.   | 65  |
| 3.17 | Other boxel shapes. . . . .  | 67  |
| 3.18 | Minimum U-Turn radius as a function of edge length. . . . .  | 68  |
| 3.19 | The position of a hit is independent of the gradient strength. . . . .   | 70  |
| 3.20 | A Jordan curve separates inside from outside. . . . .  | 72  |
| 3.21 | Voxel definitions. . . . .   | 78  |
| 3.22 | Definitions of different types of adjacent voxels. . . . .   | 80  |
| 3.23 | Voxel faces are connected by a shared edge with a hit. . . . .   | 83  |
| 3.24 | Two disconnected connected systems of hits passing through a voxel.  | 84  |
| 3.25 | Triangle edges on voxel faces must be continuous with triangle edges<br>in the adjacent voxel opposite face (“voxel face-wise continuity rule”). | 88  |
| 3.26 | Each triangle edge must be a “two way street” (digraph) to be coher-<br>ently oriented. . . . .  | 92  |
| 3.27 | Triangle face and vertex normal vectors. . . . .   | 95  |
| 3.28 | Spinning a single triangle in a voxel face. . . . .  | 100 |
| 3.29 | Three triangles in three connected faces spins a locally connected surface.  | 102 |
| 3.30 | Four triangles in four connected faces spins a connected surface. . . .  | 104 |
| 3.31 | Four triangles in one faces spins a bubble in the surface for SpiderWeb<br>A. . . . .  | 105 |
| 3.32 | An example triangle webbing inside a voxel. . . . .  | 107 |
| 3.33 | An example triangle webbing showing triangle normals inside a voxel.   | 108 |
| 3.34 | A SpiderWeb triangles assembled in a small group of voxels. . . . .  | 109 |
| 3.35 | An example brain sulcus surface, extending deep inside the head. The<br>fissure is approximately 3 pixels (3 mm) wide. . . . .                   | 110 |

|      |   |     |
|------|---|-----|
| 3.36 | An example brain sulcus surface, showing triangle face normal vectors(“hedgehog mode”). . . . .   | 111 |
| 3.37 | A close up view of the entrance to the sulcus. . . . .  | 112 |
| 3.38 | A far view of the partial surface reconstruction of a sulcus extending below an axial MRI slice. The width of the fissure is 3 mm, the size of each pixel is 1 mm. . . . .                                      | 113 |
| 3.39 | Depth complexity and volume occupancy in a region of a data lattice.  | 117 |
| 3.40 | Blockyness Factor shifts hits to the midpoint of all edges. . . . .   | 120 |
| 3.41 | Blockyness factor set to 1.0 with triangle face normals used. . . . .   | 121 |
| 3.42 | Blockness factor set to 1.0, with triangle vertex normals used. . . . .   | 122 |
| 3.43 | Blockyness factor set to 0.5, with triangle face normals. . . . .   | 123 |
| 3.44 | Blockyness factor set to 0.5, with triangle vertex normals. . . . .   | 124 |
| 3.45 | Triangles vertices drawn in T-mesh order are drawn fastest. . . . .   | 126 |
| 3.46 | A noisy surface just below the skin about the bridge of the nose. The grey color of the surface is proportional to the magnitude of the gradient. The thick (green) lines are bubble bifurcation edges. . . . . | 128 |
| 3.47 | Eulers-Poincaré law satisfied by SpiderWeb surfaces. . . . .  | 132 |
| 3.48 | What <b>isovalue</b> best contours this geometry ? . . . . .  | 138 |
| 3.49 | The separation of contour rings by changing threshold. . . . .  | 139 |
| 3.50 | A birth voxel assembly. . . . .   | 140 |
| 3.51 | Fission/Fusion faces. . . . .   | 141 |
| 3.52 | A threshold $\tau$ / genus / object map. . . . .  | 142 |
| 3.53 | A proposed object isovalue <i>vs</i> area and volume map. . . . .   | 143 |
| 3.54 | Osculating surfaces at a 4-hit face. . . . .  | 144 |
| 4.1  | Indistinguishable isosurfaces. . . . .  | 167 |
| 5.1  | Neuromagnetometer measurement of the brain’s magnetic field. . . .  | 176 |
| 5.2  | Computer model of a Pointer, viewed from the side. . . . .  | 191 |
| 5.3  | Computer model of the Pointers, viewed from the top. . . . .  | 192 |
| 5.4  | A Pointer worn over the right periauricular point. . . . .  | 193 |



|      |   |     |
|------|---|-----|
| 5.5  | A landmark as the lowest point on the ridge profile at the saddle . . .   | 210 |
| 5.6  | Slop Distance Error. . . . .  | 218 |
| 5.7  | Warpage Percent Error is the difference in the lengths of the similarly styled lines. . . . .   | 220 |
| 5.8  | Wobble Distance Error. . . . .  | 221 |
| 5.9  | A representative axial MRI slice, showing Pointers traces. . . . .  | 226 |
| 5.10 | The Blobulator: magnified blob borders and the blob weighted center of gravity. . . . .   | 227 |
| 5.11 | The appearance of nasion and the blobs of two spokes of a Pointer, in an sagittal MRI slice. . . . .  | 229 |
| 5.12 | The intersecting planes of the MRI slices and the Pointer contrast media chambers. The resultant pattern of dots as it appears in the MRI slice plane is shown for each intersection. . . . . | 231 |
| 5.13 | The Pointers intersection from vertical slices, with the resultant pattern of dots. . . . .   | 232 |
| 5.14 | CoordAligner software panel showing line fitting operation through the blobby pointer traces. . . . .   | 233 |
| 5.15 | CoordAligner locating nasion from best lines through nasion Pointers contrast media chambers. . . . .   | 234 |
| 5.16 | CoordAligner locating cardinal landmark from Pointers over right periauricular point. . . . .   | 235 |
| 5.17 | Pointer traces in a sequence of sagittal a slices through nasion. . . . .   | 236 |
| 5.18 | Pointer traces in a sequence of sagittal a slices over a periauricular point. Distance between axis ticks are 1 cm. . . . .   | 237 |
| 5.19 | A coordinate system before alignment to cardinal landmarks. . . . .   | 238 |
| 5.20 | After registration, the slices are tipped into the head-frame coordinate system. Two slices are shown, the right slice passing through the nasion   | 239 |
| 6.1  | Subject SW nasion photograph, profile view. . . . .   | 268 |
| 6.2  | SW nasion as rendered from a coronal stack with the SpiderWeb algorithm. . . . .  | 269 |
| 6.3  | SW nasion as rendered from a sagittal stack with the SpiderWeb algorithm. . . . .   | 270 |

|      |  |     |
|------|--|-----|
| 6.4  | SW nasion, another view, as rendered from a sagittal stack with the SpiderWeb algorithm. . . . .                               | 271 |
| 6.5  | Subject SW left ear photograph. . . . .  | 272 |
| 6.6  | Subject SW left ear reconstruction from sagittal slices with left periauricular point marked. . . . .                          | 273 |
| 6.7  | ZL's Frontal Face Photograph. . . . .  | 274 |
| 6.8  | ZL Nasion Photograph. . . . .  | 275 |
| 6.9  | Subject ZL nasion reconstruction from a coronal stack. . . . .   | 276 |
| 6.10 | Subject ZL nasion reconstructed from sagittal slices. . . . .  | 277 |
| 8.1  | The best sphere that fits into the occipital pole of the brain case. . .   | 291 |
| 8.2  | Voronoi diagram and Delaunay Triangularization in the plane. . . . .   | 295 |
| 8.3  | Best sphere and SQUID coils before Delaunay triangularization. . . . .   | 296 |
| 8.4  | Side view best sphere and SQUID coils before Delaunay triangularization.   | 297 |
| 8.5  | Voronoi surface of SQUID coil pickup. . . . .  | 298 |
| 8.6  | Delaunay triangular tileing of the SQUID coil pickups. . . . .   | 299 |
| 8.7  | View from behind the head, looking toward the nose of the magnetic field pattern over the auditory cortex of the head. . . . . | 300 |
| 8.8  | Lateral view of the magnetic field pattern from auditory cortex. . . . .   | 301 |
| 8.9  | A simulation of the surface magnetic field. . . . .  | 303 |
| 8.10 | CSS Inverse solution the bottom plane. . . . .   | 312 |
| 9.1  | Subject SW, Right Hemisphere Alphas, viewed over a slice approximately 3 mm Right of the midsagittal plane. . . . .            | 325 |
| 9.2  | Subject SW, left hemisphere alphas. . . . .  | 327 |
| 9.3  | SpiderWeb surface of the parieto-occipital sulcus about an alphon. . .   | 328 |





# List of Tables

|      |  |     |
|------|--|-----|
| 6.1  | Overview of registration studies from acceptable MRI stacks. . . . .   | 254 |
| 6.2  | Cardinal landmarks for subject SW, as measured by ZL, using a Polhemus probe, with 3 repetitions of 3 measurements procedures. Measurements in cm. . . . .                             | 254 |
| 6.3  | Cardinal landmarks for subject ZL as measured by SW, using a Polhemus probe, with 3 repetitions in each of 2 procedures and 2 repetitions for 1 procedure. Measurements in cm. . . . . | 254 |
| 6.4  | Cardinal landmarks for subject DK measured by ZL, using a Polhemus probe, with 4 repetitions in each of 2 procedures. Measurements in cm.  | 255 |
| 6.5  | Cardinal landmarks for subject ZL, measured by DK, using a Polhemus probe, with 5 repetitions in 1 procedure, and 4 repetitions in 1 procedure. Measurements in cm. . . . .            | 255 |
| 6.6  | MRI Studies on the Philips Gyroscan 1.5 T system, with pixel and slice sizes expressed in cm. . . . .  | 256 |
| 6.7  | Direct Localization Coordinates of Cardinal Landmarks expressed in PPN Headframe. Measurements in cm. . . . .  | 258 |
| 6.8  | Direct localization <i>vs.</i> Polhemus digitizer probe measurement error distances in cm. . . . .   | 260 |
| 6.9  | Direct method percentage change in distances between measurement modalities. . . . .   | 262 |
| 6.10 | Pointer Localization Coordinates of Cardinal Landmarks. . . . .  | 264 |
| 6.11 | Pointer localization <i>vs</i> Polhemus digitization measurement error distances in cm. . . . .  | 265 |
| 6.12 | Pointer technique percentage change in distances between measurement modalities. . . . .   | 266 |

|      |   |     |
|------|---|-----|
| 6.13 | SpiderWeb Localization Coordinates of Cardinal Landmarks. Values in cm. . . . .                                     | 279 |
| 6.14 | SpiderWeb localization <i>vs.</i> Polhemus digitization measurement error distances. Distances in cm. . . . .       | 280 |
| 6.15 | SpiderWeb technique percentage change in distances <i>vs.</i> Polhemus probe measurements. Distances in cm. . . . . | 281 |
| 7.1  | Summary Error Slop, Warpage, and Wobble compared across techniques.   | 285 |
| 9.1  | Stable (stationary) Alphons recorded. . . . .   | 320 |
| 9.2  | Strong Alphons . . . . .  | 321 |
| 9.3  | Medium Alphons. . . . .   | 322 |
| 9.4  | Weak Alphons. . . . .   | 323 |
| 9.5  | Right hemisphere alphons ranked by perpendicular distance from nearest sulcus. Subject SW. . . . .                  | 324 |
| 9.6  | Left alphons ranked by perpendicular distance from nearest sulcus. Subject SW. . . . .                              | 326 |

# **Part I**

## **Introduction to Dissertation**



The goal of this dissertation is to project measurements of the brain's magnetic field onto maps of brain structure. Then we can determine what part of the brain's activity produced the observed magnetic field. The main problem is that a different measurement system was used to collect brain magnetic field signals and to collect brain anatomic structure from Magnetic Resonance Imaging. I needed to develop techniques to combine these disparate coordinates into a common, head-based coordinate system.

The research projects described in this dissertation focus on issues in isodensity surface construction from volume density data and registering these data into a common coordinate system. "SpiderWeb" isodensity surfaces were constructed in noisy Magnetic Resonance Image (MRI) stacks of slices. A common head based coordinate system is used to combine both Magnetic Resonance Imaging and Magnetic Source Imaging (MSI). Finally, this research maps functional signals from the brain onto the anatomic geometry obtained from MRI.

Our Magnetic Source Imaging (MSI) models the anatomic geometry of sheets of normally oriented electric current dipoles in the electrically active cortex of the brain, recorded across a measurement surface above the scalp. Problems in the generation of a consistent surface from noisy MRI data prompted the conceptualization and development of the "SpiderWeb" algorithm. Problems with localization of Cardinal Landmarks resulted in the invention of the Pointer Hardware and development of the "Best Point and Line" algorithm. Problems in relating a stack of MRI slices to MSI locations motivated the development of the "CoordAligner" software system to re-orient MRI stacks into a common head based coordinate system. This allows interactive investigation of magnetic signals in MRI slices. An investigation of local brain alpha spindle activity, or "alphon" was carried out to relate the anatomic structures that may be responsible for these signals.



## **Part II**

# **The SpiderWeb Isosurface Algorithm**





# Chapter 1

## Introduction to the SpiderWeb Algorithm

The SpiderWeb algorithm consists of techniques to render consistent and complete isovalue contour lines and surfaces. An example use includes building brain or skin surfaces from an Magnetic Resonance Imaging (MRI) study the head. This algorithm is simple and robust in that it builds valid surfaces in volume data sets with excessive noise or weak boundary gradients. More exactly, a SpiderWeb surface is an oriented and unbounded Jordan surface. A Jordan surface segments an inside region relative to an outside region. The output of the SpiderWeb algorithm consists of contour line segments or triangle meshes. A remarkable property of this algorithm is that there are no constraints on the datum value of an individual pixel, and yet the resultant isosurface has a number of important global topological properties. The two techniques are called SpiderWeb A and SpiderWeb B. Triangles and contour lines are constructed by a simple, cyclic “spinning” rule that knits together line segments (edges). They differ in the assumption made when spinning triangles in ambiguous situations in noisy or weak data gradients. We will develop the ideas of the algorithm first in two dimensions to spin contour lines and then in three dimensions to spin triangle mesh.

It is important that the rendered surfaces have a correct topology. Holes, tears, bridges to otherwise disconnected surfaces, and incorrect triangularization are not apparent when viewing a three dimensional reconstruction consisting of a half million triangles. These errors cause more serious errors when attempting to make quantitative measurements such as a surface area and volume. A triangle surface mesh

is required for modeling physical properties by finite element analysis[1] or boundary element analysis[2]. Volume rendering algorithms do not produce geometry elements for physical modeling. This deficiency motivated my research into surfaces and their construction in volume data.

## 1.1 Why Another Algorithm ?

The SpiderWeb algorithm solves the major problems encountered in other popular isosurface algorithms. Additionally, I feel that this solution is elegant. The surface is built using only local voxel information knit together with a simple, cyclic triangle spinning rule. From only this local voxel information, I derive the global topological properties, and show that these properties are appropriate and correct for modeling and visualization studies.

Analysis of other algorithms, described in Section 2, reveal that issues in topology were frequently overlooked in the quest for fast rendering speed. In many algorithms, subtle tileing errors are lost in a blizzard of small triangles. In those cases where fast visual rendition are the only consideration (such as Lorensen and Cline's [3] "Dividing Cubes" algorithm), one can not complain about topology. However, where a critical definition of a surface is required, some algorithms are not specified in the literature with enough details to exactly replicate the result shown by the authors. The many problems with surface construction algorithms are in the literature will be reviewed. I do not maintain that all, or any, other algorithms are "wrong". The simple solution described in this thesis appears to have been overlooked, or possibly inadequately described, in the literature. Additionally, many computer graphic visualization programs are plagued by problems in implementing this class of algorithm. Even if the correct algorithm has been described in the literature, it has not been communicated to the professional programmer in a way that has resulted in error free implementations. The novel algorithms disclosed here are valueless to the computer graphics community unless I can describe them in a way that will enable a professional scientific programmer to implement them directly and unambiguously from our instructions.

Surface tileing errors are widespread in the literature, and a representative example is shown in Figures 1.1 and 1.2. We can see in these figures that there are a number of triangles missing that would otherwise make a continuous, hole free surface. I show in Section 3.2.2.2 that in order to generate a continuous surface tileing, there



are many more individual tiling cases to be considered, involving each adjacent voxel. However, the SpiderWeb algorithm does not use any case tables. All triangles are generated by a triangle “spinning” rule consistently applied in all cases.

I demonstrate and describe how to use the algorithm rules for building SpiderWeb surfaces simply, precisely, and unambiguously. These rules are sufficiently general to give correct results for any space filling voxel geometry (tetrahedrons, pie wedge fillets, hexagon bee hives, *etc.*) without tetrahedral or simplicial complex decomposition. These rules can be formalized and generalized for higher dimensionality voxels, such as time voxels (TiXels [4]). The generalized problem of matching contour rings drawn or calculated in a slice is solved by the inter-slice gradient information [5]. We show that tiling triangles inside a triangle is basically a cyclic knitting of connected isovalue threshold crossing (hits) along voxel edges. These knitting rules can be compiled into a bitmap encoded jump table, or can be coded in “C” as a few tight computer instruction loops. The SpiderWeb A surface responds to weak or noisy gradients with the generation of small non-manifold bubbles. Extended noisy regions that produce multiple adjacent bubbles generate surfaces with ambiguous internal regions that resemble closed cell foam rubber. These internal triangles can be scraped out of the inside of foamy regions. A measure of the ambiguity in a noisy region can be ascertained by comparing measurements of the inside-side surface with the outside-side surface of a foamy region. The differences between these surfaces can provide error bars for noisy volume and surface measures. In addition, the SpiderWeb algorithm surface can be displayed with the magnitude of the gradient, or strength of the surface. This overcomes the illusion generally propagated in isosurface renderings where the isosurface is displayed as an absolute, “brick wall” boundary [6].

### 1.1.1 Novel Features of SpiderWeb Algorithm

The main feature of this algorithm that sets it apart from other algorithms, is that it generates correct surfaces despite random noise fluctuations in the volume density values. Some other algorithms such as implementations of the marching cubes algorithms develop holes or tears when the volume data values are noisy.

We will show that the SpiderWeb isosurface does not have artifactual holes under any possible combination of data voxel vertex values.

We will show that a SpiderWeb surface will always meet up with itself. There are no loose ends of the surface, unless a cutting plane or a volume of interest

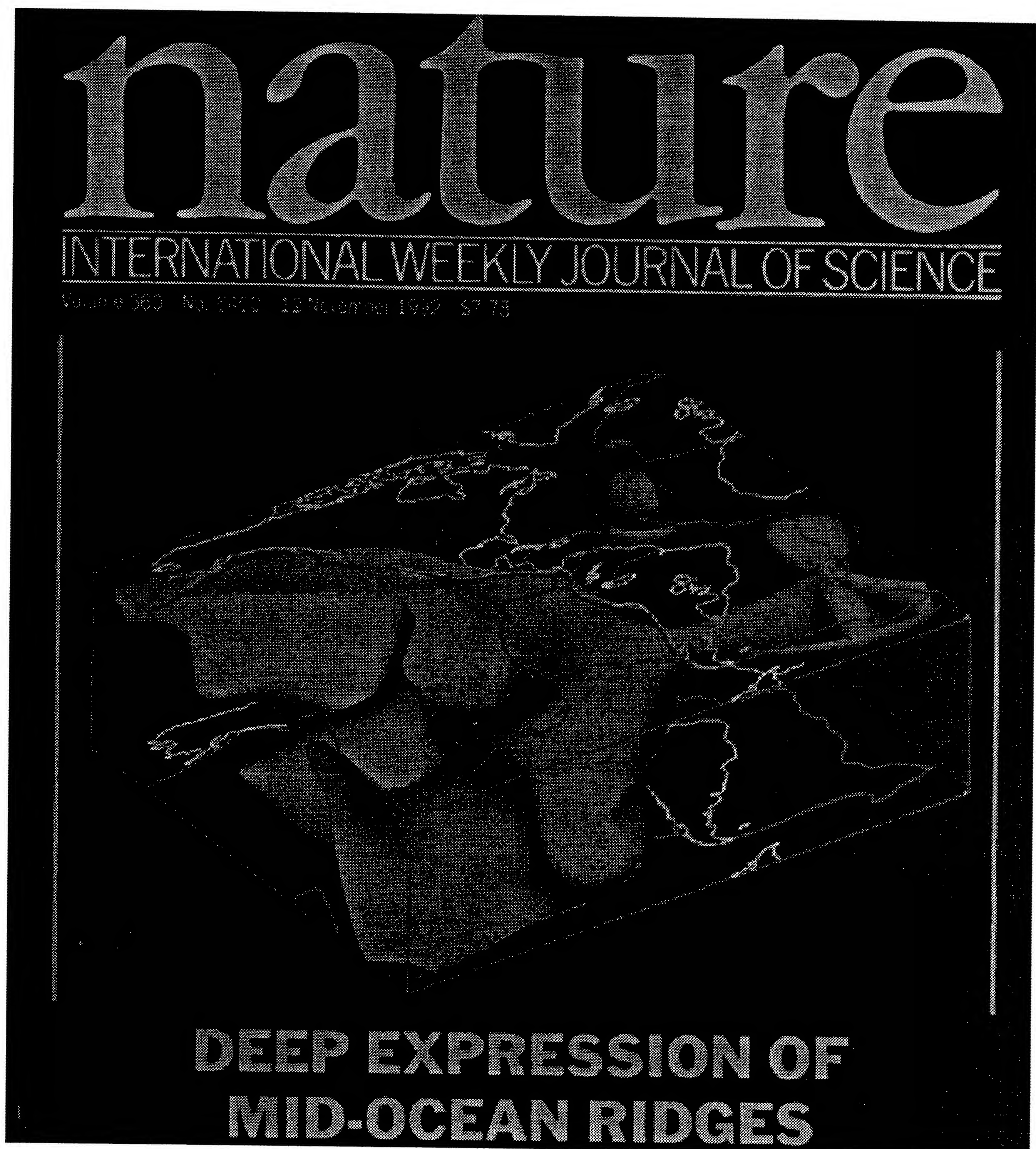


Figure 1.1: An example of surface tileing errors in a prominent place in the literature.





Figure 1.2: Detail of surface tiling errors in preceding Figure.



box is specified to limit the growth of the surface.

The SpiderWeb is a depth-first recursive tracking algorithm. It tracks a surface to the exclusion of other, possibly nearby adjacent surfaces. This makes it useful in noisy environments. It is noise ignoring. An algorithm that marches through all voxels in a volume of interest will render all possible surfaces, which can cause noisy isovalues to obscure the object we wish to view. Normally, one would filter the volume data to reduce random, disconnected noise prior to rendering a surface. Volume filtering is not required for a good SpiderWeb surface.

Once the proper data structures are constructed, any possible isovalue surface can be simply constructed. The essential operation is triangle construction (spinning) by consistent voxel circulation about threshold crossings (hits).

The SpiderWeb recursively builds out from a seed voxel a connected sheet. By iterating through each voxel as a possible seed for recursion, we can count disconnected objects, or effectively implement the algorithm as an iterative “marching” type algorithm.

Adaptive gridding algorithms used in finite element methods and boundary element methods can generate geometry element of different shapes. Fillets are best modeled by wedge shaped geometry elements, bulk solids by cuboid elements, and other elements determined by the meshing algorithm. We show that there is no need to do a (potentially incorrect) decomposition into tetrahedra or triangles.

### 1.1.2 Example Medical Illustrations

The main problem we face in three dimensional reconstruction is how to go from a planar sampled image, such as the example shown in Figure 1.3 to a three dimensional surface model of an object as shown in Figure 1.7.

Figure 1.3 is a sagittal slice through the head taken by the Philips S1.5 GYROSCAN<sup>TM</sup> MRI system at New York University Medical Center. The sample image consists of 512 by 512 pixels in the image plane. The dynamic range of the pixels is from 0 to 4095 (or  $2^{12} - 1$ ), where 0 is black and 4095 is white. The value of a pixel represents, when scaled, microvolts of magnetic resonance imaging echo signal returned by the tissue at each  $z, x$  location in the slice plane  $y$ . Also visible above

the nose are the spokes of the Pointers registration devices, which are the subject of the Section 5 of this dissertation.

A problem in medical imaging is that the dynamic range of the computer graphics workstation display and the printers is at most 8 bits of Red, Green, and Blue. We need to capture all the information that is in the medical image, and not reduce the dynamic range of the data to fit the display technology. We would lose important fine detail in that way.

Figure 1.4 shows a new technique for rendering 12 bits of dynamic range data into 8 bits of display capability by making a *bas* relief of the image. The pixel tiles of the image plane are smoothly elevated by the value of the pixels. From the raised tiles, three dimensional surface normals are calculated. The light from a low angle light source grazes the surface to reveal small features in the density as artificial bumps and groves. The sphere in the upper right hand corner signifies a source of light. The image we see is the interaction of the light with the surface, the surface texture, and the surface orientation (normals). The low level noise at the border of the skin is visible as a crinkled texture in the image plane. These crinkles have a range of only 3 to 4, but they interact with specular reflectance from the lighting source to give a rich texture that would be washed out in an 8 bit grey scale imaging. This thesis was printed at 600 dots per inch (dpi) with effective 85 dpi grey scale pixel resolution [7]. The number of dithered grey values are 50 ( $2^5 = 64$ ). So you, the reader, are looking at 14 values less than full 5 bits of grey scale resolution. The *bas* relief technique brings out small variations in image value by representing them as a surface texture, and reveals detail that might be otherwise lost by limitations in the printing process. However, the *bas* relief technique is not a true three dimensional reconstruction.

Figure 1.5 shows a series of two dimensional slices stacked up in space. The outside noise has been trimmed or thresholded away, and we see the border about the head instead of the slice rectangle. This is not a three dimensional reconstruction, but a representation of the slices as they intersect the actual three dimensional object.

Figure 1.6 shows the full, three dimensional rendering of the SpiderWeb skin isosurface of the head (of a different subject). We can see MRI artifacts and noise from a clinical high resolution study. Noise artifacts are visible about the eye, and a wrap around of the bottom slices to the top slices is visible at the crown of the head. Figure 1.7 shows detail from the skin surface about the brow, more precisely, the cardinal landmark called nasion, used to construct the head-based coordinate system. Figure 1.8 shows detail about the ear. The sharp curvature of the pinna is





Figure 1.3: A representative sagittal MRI slice of the head.





Figure 1.4: A sagittal MRI slice, shown as a *bas* relief.



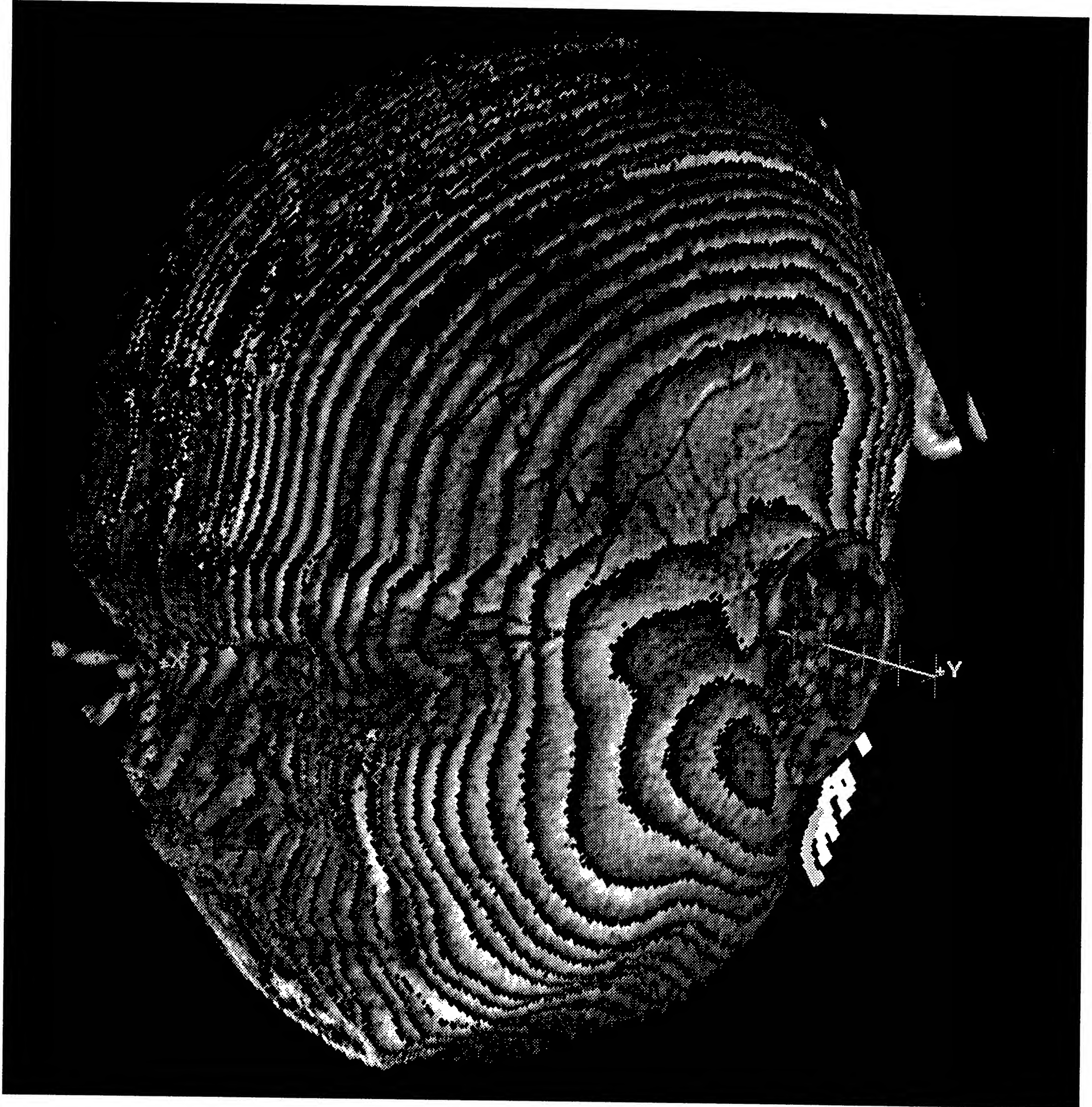


Figure 1.5: A stack of trimmed sagittal slices of the head.



smoothly demonstrated. The bottom of the ear was out of the field of view of the MRI study.

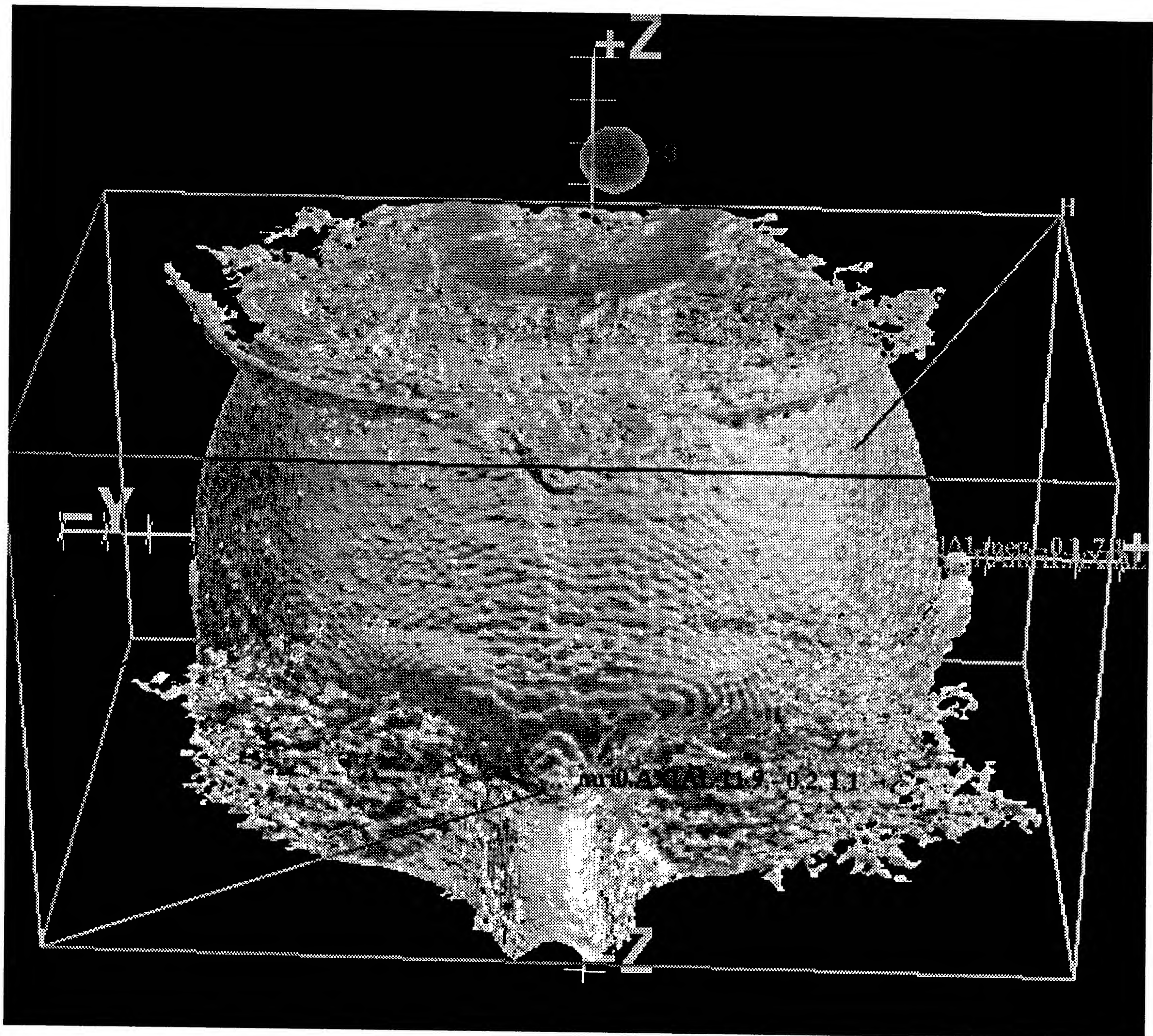


Figure 1.6: The skin isosurface of the head reconstructed from a stack of MRI slices.



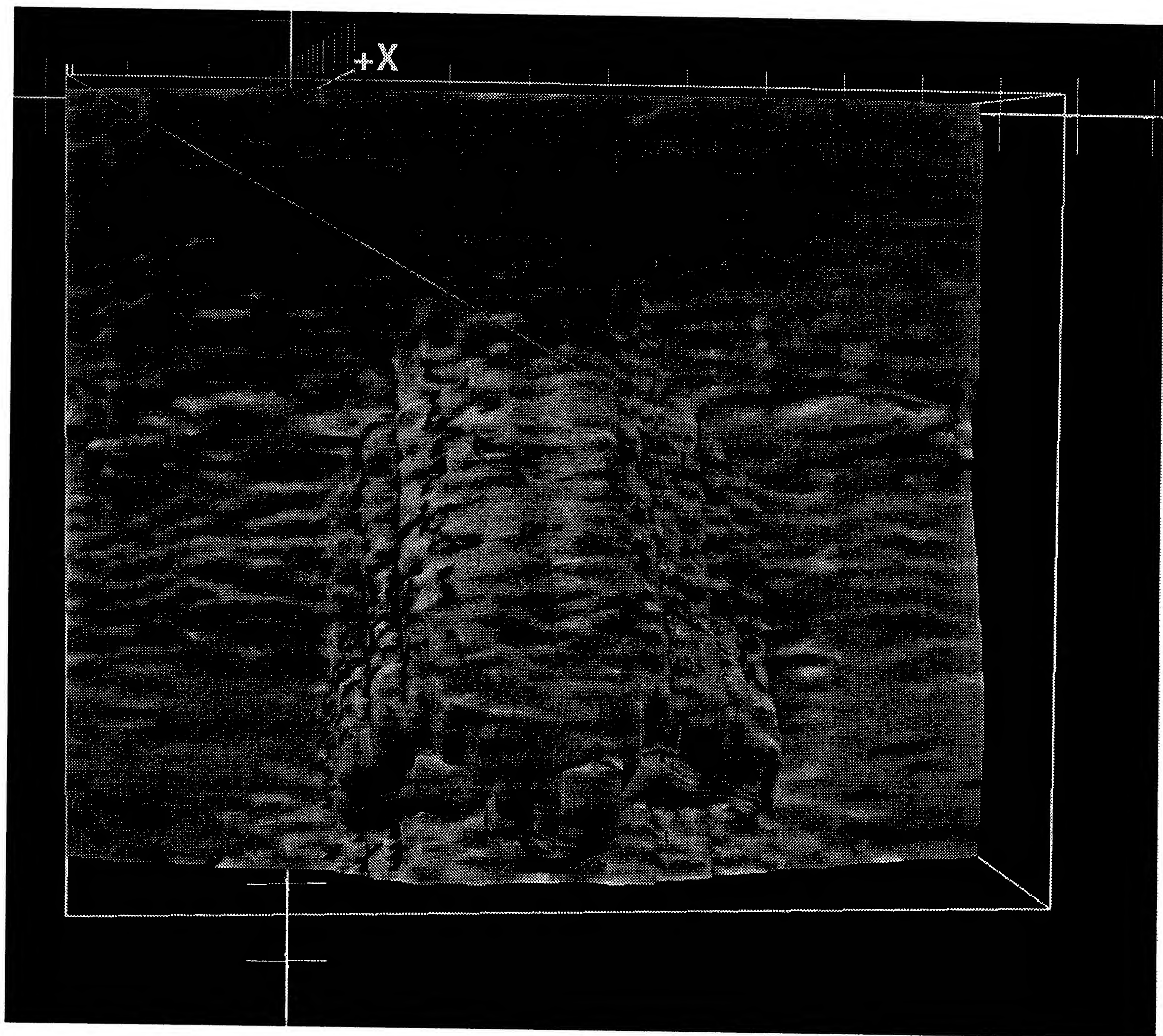


Figure 1.7: The skin isosurface detail of the reconstructed nose and brow.



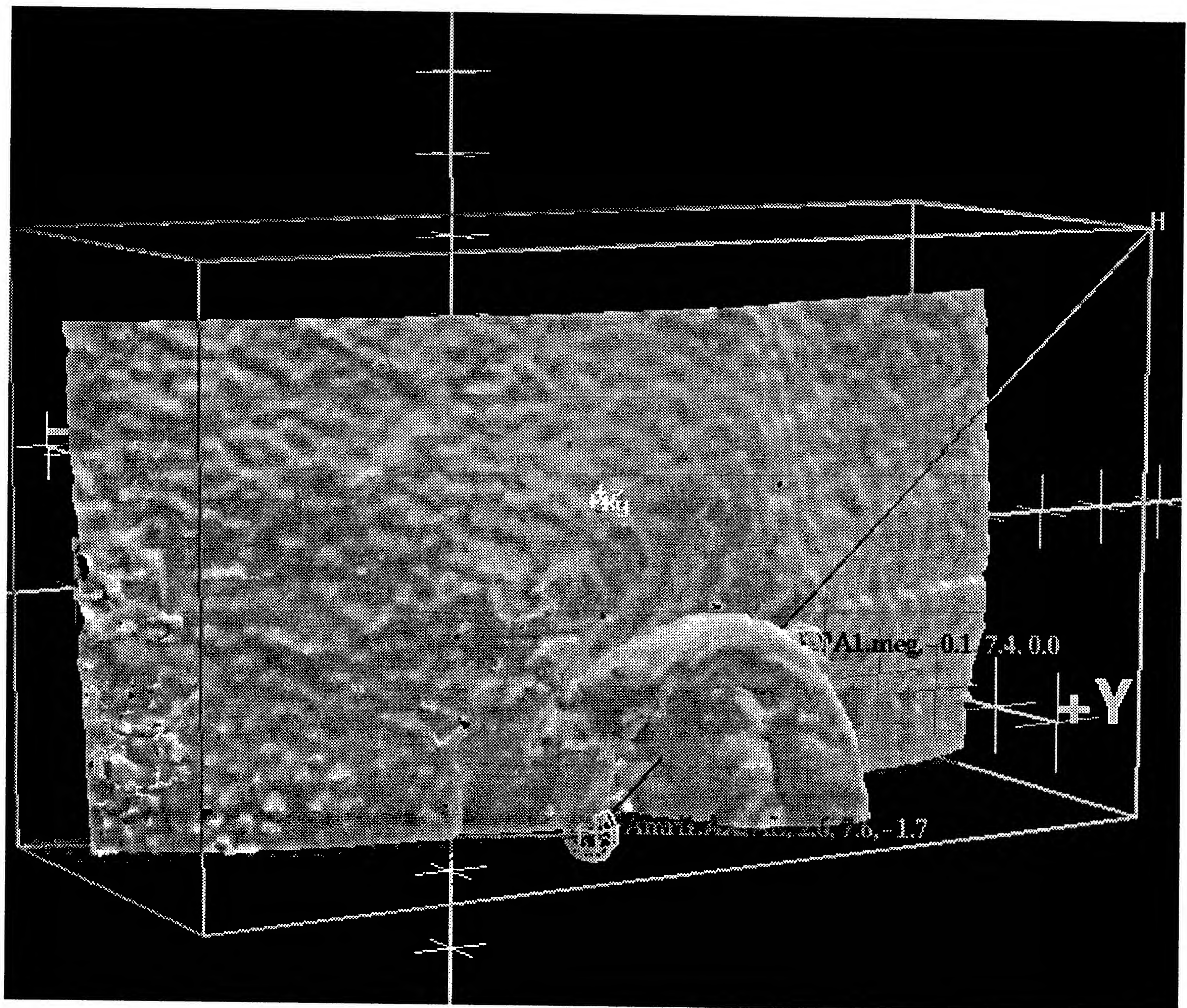


Figure 1.8: Skin isosurface detail of the ear.



A feature of the SpiderWeb algorithm is that it builds valid surfaces in the presence of noise. Figure 1.9 shows the results of a planar volume gradient thresholded halfway through the gradient. A few slices in the gradient have had noise added to show the response of the surface to noise. The noise causes the otherwise smooth surface to writhe and distort. A control panel button permits the continued injection of noise to a slice to interactively demonstrate the coherence and correctness of the surface.

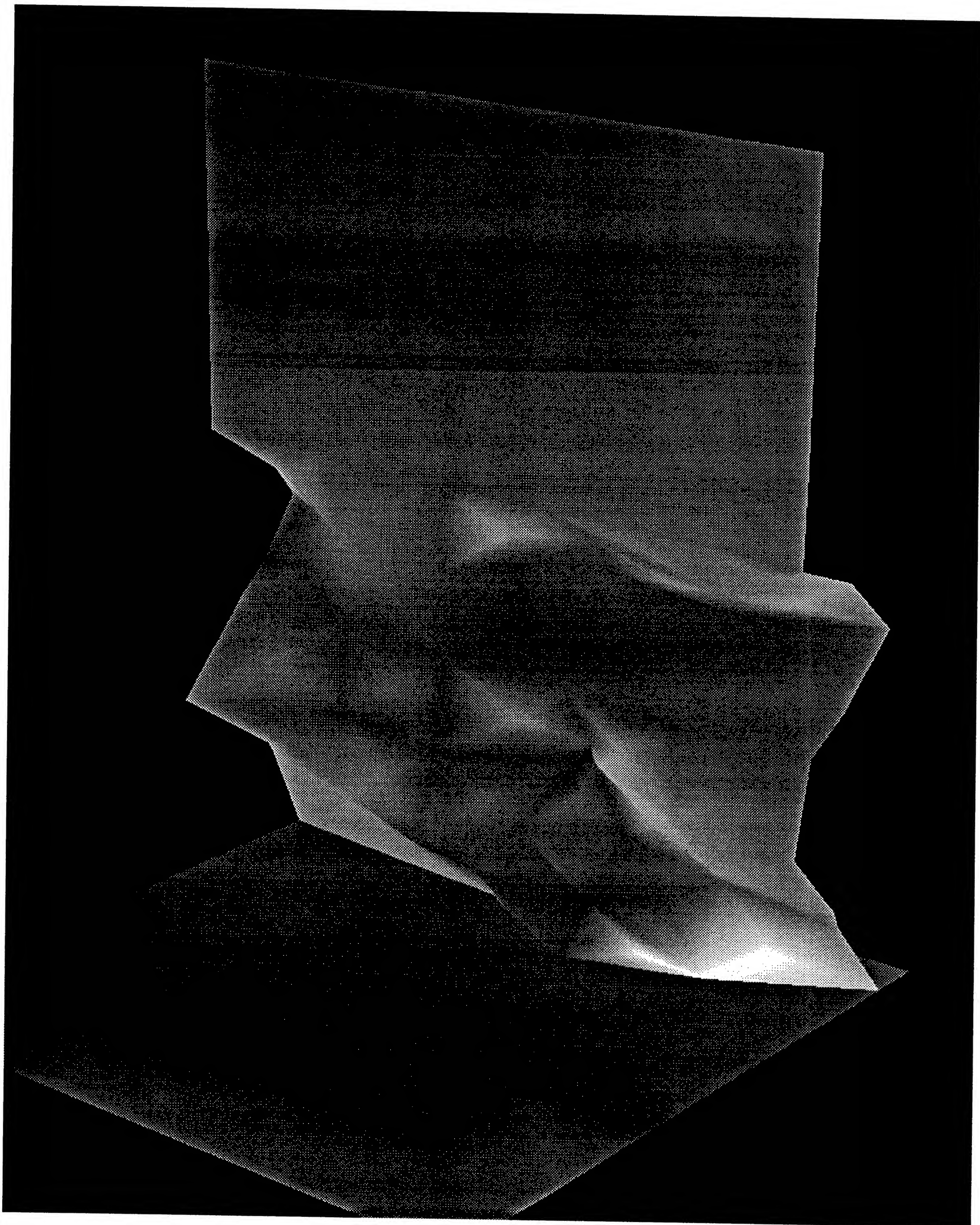


Figure 1.9: An artificial linear volume gradient with a few slices perturbed by random noise.



### **1.1.3 Physical Measurements Confirm Fidelity**

Measurements made by a Polhemus Navigation, Inc., [8] three dimensional digitizing probe of skin landmarks on the head are shown in this dissertation to agree with measurements made by on SpiderWeb surfaces to within approximately 1.2 mm, depending on the experience of the measurer. This correspondence between physical measurements and measurements computed on the SpiderWeb surface indicates that we can have confidence in the geometric fidelity of the algorithm. The geometric fidelity is limited by the calibration and quality of the basic measurement system used. It is also operator dependant, in the sense that the physical sighting of the skin landmarks accurately and precisely is a learned skill. We show in this dissertation that the SpiderWeb algorithm, in addition to topological correctness, also has a sufficient degree of geometric fidelity to permit critical dimensional measurements.

## Chapter 2

# Critical Review of the Literature

There are two principal approaches to apprehending volume density data, which we term volume visualization and surface rendering. In Volume Visualization, a row or column of volume elements (voxels) is traversed and mapped by an opacity function to a screen pixel. Surface Rendering produces a geometric definition of an object for further analysis. These two divergent approaches to apprehending volume information are discussed below, and I show that the two schools of thought are converging. Stytz [9] *et al.* 1991 provides an extremely comprehensive, albeit dated, review of the medical imaging.

Surfaces are rendered from a volume by two main methods. The first class of surface rendering techniques involves contouring isovalue rings in slices [5] and then connecting the rings between slices. This is the classical approach used in 3D reconstruction in optical microscopy, microtomy, and other situations where the native data has a slice-like nature. The second surface rendering approach is direct three dimensional isovalue surface generation, analogous to isovalue contour lines. In this approach, triangular tiles are knit directly inside volume elements, without intermediate contour rings. In Surface Rendering, a list of triangles is generated (rendered) and visualized. This triangle mesh surface can then be used for further physical property modeling by finite element analysis (FEA) or boundary element analysis (BEA). Volume visualization does not produce geometry elements for subsequent physical modeling. This deficiency motivated my research into surfaces and their construction in volume data.

The establishment of algorithmic correctness and geometric fidelity is one of the goals in this project. I believe that the SpiderWeb algorithm solves the problems in surface construction mentioned above and should make surface rendering a robust imaging technique.

## 2.1 Other Surface Rendering Algorithms

Analysis of surface algorithms requires a definition of a surface. We consider a surface only if it satisfies a number of properties discussed in the following section. One important property is that a SpiderWeb surface is a Jordan surface. A Jordan surface is a boundary that separates an inside region and an outside region. This topic comes very close to issues in segmentation, where each pixel is assigned membership in one or more groups. The boundary between one group pixels (or voxels) and another can also be considered a surface. Segmentation and surface construction algorithms are similar in many ways. A Jordan surface also segments an inside from an outside. A true segmentation algorithm can have multiple groups besides inside and outside. An example from MRI would be brain, air, bone, cerebral spinal fluid, blood, *etc.*

### 2.1.1 Surface Rendering by Contour Rings

The earliest medical imaging surface reconstruction algorithms that produced true surfaces were based on contour rings. Keppel [10] in 1975 described an algorithm in which closest neighbor rings are knit together. Snyder [11] described a generalized contour line extraction algorithm in 1978. Fuchs et al. [12] describes an optimal surface reconstruction in 1977. Movie BYU [13] [14] was perhaps the first surface reconstruction software package that was based on sewing triangles between contour rings on adjacent slices. The vertices of a homologous pair of contour rings were resampled so that there were pairs of homologous points evenly distributed about each ring. A strip of triangles were sewn zig-zag between the planes of the ring pairs. This approach was not perfect. Occasionally, depending on the object being reconstructed, the program would get lost and ask the operator incomprehensible questions about how to knit things together. An early commercial venture, Contour Medical Systems, Inc., now CEMAX Medical Systems, Inc., reconstructed CT slices by operator assisted contour rings [15] [16]. Phoenix Data Systems, Inc., attempted to market a solid modeling system to medical planning, based on highly specialized



and expensive hardware [17]. The commercially available algorithms for object reconstruction have exploded in diversity, generally into volume approaches and surface techniques [18]. Newer techniques in surface construction from planar contour rings include Zyda *et al.* 1978 [19], who described an expanded triangulation algorithm that deals with multiple contour rings per plane as well as partial contours. It too requires human intervention when it gets lost. Surprisingly, slice contouring methods are still a hot research topic, as evidenced by the review by Meyers, Skinner, and Sloan in 1992 [5], and by Hall in 1990 [20]. Feldman [21] describes an interesting isovalue contour algorithm that works in pixel map data. It has many similarities to cuberille smoothed voxel algorithms. Newer, contour line propagation algorithms that use dynamic programming and neural net heuristics are under development by Morse *et al.* [22] and Udupa [23] [24]. Essentially, a “hot wire” travels as far as it can in a direction in the slice plane. The algorithm learns from the user interaction so that on subsequent probings, it reduces user interaction to guide it around a slice. Recent work by Shinagawa *et al.* attempt to provide a theoretical framework for knitting contour rings in ambiguous situations by using Morse theory [25] and Reeb graphs [26].

A strong motivation for surface rendering algorithms is the extraction of bone or heart surfaces for simulating medical procedures. Computer aided surgery is still a promise unfulfilled, but early surgical simulation ventures, hardware and software systems bloomed in the literature [27] [15] [28] [29] [30] [31] [32] [33], but not so much in the marketplace. Newer research has developed faster algorithms which include a personal computer based system by Udupa *et al.* [34], and much faster multi-processor computers that plow through large volumes of data, using brute force algorithms. An example is the ISG platform, [35] [36] [37] which is the only volume visualization platform dedicated for the medical market.

The integration of 3D reconstruction systems with an X-ray tomography system brings real meaning to the computer part of the term “computer aided tomogram” (CAT) scanning. Jackèl and Strasser (1988) [38] describe the PARCUM II system for reconstructing solids directly from raw tomogram data. Other interesting work by Thirion [39] uses the raw tomograph image, or sinogram. He segments the tissue times prior to mapping the sinogram into a slice plane. Indeed a strong motivation for 3D reconstruction from planar slices is due to the explosion of tomographic imagery from X-ray CT [40].

The blossoming of these types of systems in hospitals and doctors offices will come in the after the next generation of computer systems, where the equivalent of today's supercomputer will sit inside a CRT display pedestal. Real time volumetric

rendering systems are now being built by the Canadian company, ISG, Inc., where the use of up to 40 RISC processor chips in parallel permits a medium scale ( $512^3$ ) volume to be cut, rotated, illuminated, and viewed almost in real time [35] [37]. The volume scene is capable of being refreshed at rates of 10 times per second.

### 2.1.2 Volume Visualization Approaches

Herman and Liu set the foundation for boxy voxel surface renditions in their 1979 paper [41]. They use the term *cuberille* to represent a point sampled three dimensional voxel to do “non-invasive vivisection”. They smooth the jagged *cuberille* surface boundary with various low pass filters, Gouraud’s smoothing algorithm [42], lighting reflectance models, and other techniques [43]. Artizy *et al.* discussed issues in surface detection for geometrical rendering in 1981. This paper [44] describes an early algorithm that tracks a boundary surface as a system of connected *cuberille* voxel faces. This algorithm takes the problem of surface tracking and recasts it into a directed graph traversal, in which the nodes of the graphs represent surface voxel faces. A number of papers deal with how to take an essentially corrugated, *cuberille* surface geometry and give it smooth light reflectance properties. Gordon and Reynolds (1985) [45] used depth and gradient shading to improve the appearance of a smooth three dimensional surface. A technique of gradient interpolation to smooth voxel joints is discussed by Bright and Laflin (1986) [46]. A real time discrete gradient shading model for recovering a smooth surface gradient from the *cuberille* voxel representation is discussed by Cohen *et al.* in 1990 [47]. Kaufman (1987) [48] describes a hardware implementation (The CUBE Architecture [49], which is a voxel based 3D rendering system) in which a *cuberille* voxel is smoothed and rendered at scan-conversion time for smooth apparent surface and fast rendering. The use of ray casting is discussed by Höhne *et al.* (1990) [50] in which they obtain vivid, lifelike rendering from voxels by using the gray level volume gradient in a lighting model. It is part of a generalized ray casting visualization package on a special hardware board to achieve renderings in 5 to 60 seconds. Webber (1990) [51] presents a biquadratic function for accurately approximating a surface from the local neighborhood of data lattice vertex densities. This smoothed surface patch permits accurate shading calculations for ray tracing using the surface normal derived from the biquadratic patch. Thus volume imaging seems to have come around to surface rendering in order to get a realistic image by estimating an interpolation surface for light reflectance modeling.

The use of *cuberille* voxels for surface measurements was demonstrated by Hemmy and Tessier in 1985 [52]. A comparative analysis of volume and surface



rendering was done by Rusinek *et al.* in 1991 [53]. They rendered a dried skull high resolution CT scan with both the “ALLIGATOR” [54] surface algorithm and the “ChapVolumes” software from Pixar [55]. Qualitatively, both renderings produced similar looking images, and had similar difficulty properly rendering thin bone and cranial sutures. Linear measurements of the dried skull were compared with similar linear measurements of the Alligator surface and the ChapVolume. There were no statistically significant differences between the measurements made by each rendering algorithm with respect to direct hand caliper readings. The statistically insignificant errors, averaging less than 1.5 mm, were attributed to incorrect localization of fuzzy edges in the volume images, and partial volume transparency. These qualitative studies in the physical fidelity of surface construction algorithms are confirmed in Section 6.

### 2.1.3 Surface Rendering Algorithms

Surface rendering algorithms start with the assumption that a smooth surface can be interpolated in a volume. Researchers of volume visualization algorithms split into two separate schools of thought, one based on volume visualization and the other centrally concerned with surface rendering and modeling issues. Workers using cuberille voxel volume rendering algorithms continued with the refinement of a corrugated voxel surface into a smoothed surface, and seem to be rejoining the surface school of surface rendering after diverging for a decade.

Zucker and Hummel in 1981 [56][57] proposed a computationally intensive approach to find an oriented plane segment that separates voxels. These planar segments can be used to bevel voxel edges into a smooth surface. Morgenthaler and Rosenfeld 1981 [58] sought to build a collection of surface points and form a Jordan surface. Their basic idea is to take the voxels that straddle a threshold boundary and thin the bodies of the voxels into surfaces. A number of theorems for 26-connectivity, and 6-connectivity are proved.

In 1986 Wyvill, McPheeters, and Wyvill [59] published the first paper on 3D contouring with triangles, and they made the first observation that an additional center point to the body of a voxel disambiguates certain difficult situations in fitting triangles into voxels. The properties of an additional disambiguation point were also noted by VanGelder in 1990 [60].



Two similar algorithms were published in 1987 and 1988 that represent true three dimensional smooth surface interpolation. Baker [61] described the “Weaving Wall” algorithm. He used the idea of signed hits, where isovalue threshold crossings along voxel edges are either entering an interior region (+), or leaving (−).

Lorensen and Cline [62] [63] [3] [64] [64] published the “Marching Cubes” algorithm in 1987 and 1988. This approach is based on their observation that the 255 possible threshold crossings on the 12 edges of a cube could be reduced by three axis rotational symmetry to 14 cases. Additionally, each triangle is shaded with the volume gradient at the triangle vertices. The Lorensen and Cline [62] hallmark 1987 paper was a significant improvement over previous 3D algorithms, because the size of a voxel limited the resolution of 3D reconstructions. This results in images that appear block-like or corrugated compared to the smooth tissue surfaces they represent. Marching Cubes simply (although when initially described, incorrectly) interpolates surface contours in a volume with triangles. The point brought out in the foregoing literature review is that the cuberille voxel approach attempts to produce the illusion of smoother surfaces. But smoothing the essentially corrugated nature of the surface blurs out important detail at the sub-voxel level. This paper gave an approach to building a sub-voxel surface, and has proved very popular in the computer graphics community. However, the algorithm as originally stated, has subtle flaws. Durst noted tiling errors that caused holes in Marching Cubes surfaces [65]. Subsequent versions of Marching Cubes repaired these holes, by considering complementary cases to the 14 canonical cases, the algorithm is considered corrected. However, many implementations of marching cubes [66] continue to have tiling flaws that need more research into the topological theory of isosurface construction in the discrete domain, similar to work done by Udupa and Herman [44] [67] [68] [69] in cuberille voxel surfaces. The “Dividing Cubes” algorithm [3] avoids the issues of geometric representation completely by subdividing the voxel cuboid into points with normals that render to exactly one screen pixel, bypassing the knitting of points into edges and triangles. This eliminates problems with small triangles, as the scan conversion to pixels is done directly at the voxel level, instead of going through an intermediate polygon conversion, then scan converting these polygons into screen pixels. This rendering algorithm is extremely fast and popular. However, there has been scant interest in topological issues in Marching Cubes, with the main attitude being that these are such rare flaws that they are not noticeable in a visualization [70], with an impact on the programming community similar to floating-point round-off error [71]. I believe that these flaws are evidence of topological changes at “nearby” threshold values.

Bloomenthal studied polygonization of implicit function surfaces in his

paper in 1988 [72] [73]. His main approach was to use implicit functions and successively subdivide the voxels as needed to capture essential features of a complicated surface function. Further work by Hall and Warren in 1990 [74] [70] extend the idea of adaptive polygonalization for implicitly defined surfaces. Their goal was to find a polygon mesh that matches some surface generating function within a specified tolerance. They also reduced the number of skinny polygons. Their objective was to provide meshes for finite element analysis. Besides visualization, mesh generation is an important preprocessing step for finite elements and boundary elements [2]. In both of the adaptive algorithms mentioned previously, ambiguity is resolved by increasing sampling resolution where needed. With a given, real medical data set, this is impossible. At some point in the future, the technology of medical imaging may be sufficiently integrated with the 3D reconstruction computer to permit adaptive rescanning of ambiguous or intricate regions of anatomy. Hand held ultrasound probes used for incremental volume rendering are a step in adaptive volume scanning [75] [76]. The operator adaptively scans an anatomic region to make it clearer in 3D. The SpiderWeb algorithm defines an ambiguous voxel, which can be subdivided and rescanned at increasing resolution until the limits of instrument resolution are reached, or the ambiguity is resolved. The real issues that need to be resolved are how to effectively deal with undecidable tiling decisions when there is no possibility of getting more information [77].

Basic neuroscience research requires brain models for further analysis. Visualization, while nice for making presentations, is not the central research focus: geometric modeling of the brain is the goal. Payne and Toga [66] [78] discussed techniques for building isosurface models of the brain and the white-gray matter interface. Toga reviews the field of experimental three dimensional neuroimaging in [79]. Schwartz and Frederick [80] [81] [82] developed a method for slicing a monkey brain and contouring the sulcus to extract surfaces. They showed that the visual images represented across the brain surface have certain invariant properties if the cerebral cortex can be flattened out without surface stretching.

Neuromagnetism measurements are more informative if deduced locations of neuronal activity can be accurately registered with respect to brain anatomy, generally derived from MRI. Preliminary work points to the need for registered and accurate characterization of the cortical surface for solving the inverse neuromagnetic model [83] [84]. Projecting single dipole fits onto the brain cortex is also required for functional anatomic fusion [85]. Simple projection of single dipole sources onto slices has been described by Williamson, Lü, Karron, and Kaufman [86]. A major focus of this dissertation is on registering coordinate systems between the MRI and neuromagnetism system for common coordinates for both modalities.



The geometric problems in physical modeling have proven the more difficult to overcome. Generation of a surface and volume mesh by 48 tetrahedron decomposition was reported by Frey and Gautherie in 1991 [87] [88] [89]. This algorithm is based on a marching cubes edge intersection algorithm to build hits into surfaces. Then the surface, which surrounds empty inside voxels, is further subdivided into tetrahedra. The surface is composed of triangular faces of embedded tetrahedra. The computational load of decomposing each voxel into as many as 48 tetrahedra make the use of this algorithm on anything other than a supercomputer doubtful. A detailed analysis of planar decomposition techniques is discussed by Agaral [90]. Classical tessellation theory is discussed by Montesinos [91]. A discussion on the geometry of iso-curvature surfaces is given by Stillwell [92]. Interestingly enough, he also gives a discussion of desingularization to produce symmetric tessellations, but no application of these topological techniques is discussed.

A massively parallel computer, the CM-2 Thinking Machines Connection Machine, which has 65536 processors, has been used to run a parallelized marching cubes algorithm [93]. The operating system hides the allocation of any particular processor such that each processor can simulate any number of virtual processors. In this way, each voxel can be assigned a processor to reach a performance of 170,000 disjoint triangles per second. This is more triangles than the fastest commercially available hardware can display at this time. The fastest workstation cited was the SGI VGX with "MultiBuffers", that is rated 150,000 disjoint triangles per second. The manufacturers stated rating for a triangle mesh is almost 1,000,000 triangles per second.

The desire for computational efficiency and geometric correctness ("legitimate surface", in Udupa [69]) has produced a number of marching cubes fixes and variants. Nielson and Hamann in 1991 attempted to fix ambiguous cases by examining adjacent voxels for clues [94]. The main solution to tileing errors in Marching Cubes by other authors has been tetrahedral decomposition, where the cube is subdivided into tetrahedra [66] [95] [60], however this requires a number of trilinear and bilinear interpolations to calculate the diagonal hits, and the tetrahedralization of the cube has not been proven unique. Indeed, the lack of a unique tetrahedralization has significant implications for the fundamental ambiguity of certain hits on a cube. Tetrahedralization one way gives a different tileing from tetrahedralization another way. There is no way, as yet, to determine which tetrahedralization is correct. A proof of correctness of any particular tetrahedralization is required to further prove the tileing from that tetrahedralization valid.

Another approach to geometric coherency was taken by Kalvin in 1991

[54][96]. His basic algorithm was to build a carefully constructed case table to obtain triangles, then insert these triangles into a winged edge [97] Constructive Solid Geometry (CSG) data structure implementation by Haddad [98]. The main approach of the ALLIGATOR system to insure geometric fidelity was to load all triangles obtained from the voxel hit case lookup tables into the winged edge database by winged edge operators to maintain overall topology. A similar, but less sophisticated approach was used by Miller in 1991 [99] where triangles were built into the wall of an expanding balloon surface to represent internal body cavities.

The excessive complexity in maintaining coherency in winged edge databases [100] [101] has prompted this line of research. I use principles of geometry and topology to show if you knit the triangles correctly, from local information only, the global topology falls into consistency without filtering it through winged edge CSG operators. That problem in surface rendering operations is the large number of triangles produced. A typical triangle count is about 500,000 to 1,000,000 for a entire head with 1 mm  $256 \times 256 \times 256$  slices. This is an even worse problem with the SpiderWeb algorithm, in that it produces three co-planar triangles for three hits on the corner of a voxel. Triangle culling, both during triangle spinning and as a post processing step is clearly required for reducing the bulk of memory and speeding rendering. Schroeder [102] has developed a fast algorithm, triangle decimation, to cull almost co-planar triangles into single triangles that shows promise in this area. Vast reductions in the number of triangles, on the order of 10:1 were obtained with no visible reduction in geometric resolution. Real time fly through were possible once a half million triangles were decimated to 40,000 polygons. However, when a decimated object is brought close to the eye point, the reduction in object detail is apparent. I believe that multi-resolution modeling is the solution. This means that when the entire head in view, only a coarse model need be displayed. As it is brought closer, only a subset of a finer model need be drawn, until it is so close that the view encompasses only a small fraction of the entire, full resolution model.



### 2.1.3.1 Problems in Surface Algorithms Acknowledged in Literature

Levoy's 1991 [103] cites problems with isosurfaces:

Unfortunately, automatic fitting of geometric primitives to sample data is seldom entirely successful. Edge following algorithms, for example, occasionally go astray when processing complicated scenes, forcing the user to intervene. Algorithms for following isovalue contour surfaces have difficulty resolving ambiguities at saddle points, resulting in incorrect tileings.

In a recent article, Argiro and Van Zandt [104] commented on surface rendering:

Choosing a surface-extraction method imposes limitations on your work. Before you start the visualization step, you must decide which voxels you're interested in. You have to throw away most of the voxel data in the procedure. In addition, you may find it difficult to determine where a supposed surface cuts through the volume. It's difficult to locate surfaces when the original data is contaminated with random noise, as is often the case with real-world instruments functioning at the limits of their sensitivities. Furthermore, some voxel data (*e.g.*, a fuzzy gradient) does not lend itself to surface extraction. You can represent fuzzy data more completely with direct volume renderings.

The nagging problems in establishing a "correct" surface from volume data were surveyed by Wilhelms and Van Gelder [95]. They said in [60]:

Correct isosurface generation is important for scientists to correctly interpret their data. Our experiments show that methods that use information from neighboring cells are more likely on balance to pick correct isosurfaces, at least on the functions we tested.

It is equally important for users of isosurface algorithms to be aware of the possibility of incorrect topologies, so that they can adjust their interpretations of the data to take into account these possible errors.

The SpiderWeb algorithm produces a globally correct surface topology from local voxel information only.

Tiede *et al.* in 1990 [105] said:

...The images are more or less pretty, but experience shows that despite all their beauty they may not reflect reality. For clinical applications it is essential that one is sure about the fidelity of the pictures or at least understands its limitations.

Udupa *et al.* compared surface and volume algorithms, and concluded [106]:

In conclusion, from purely a technical viewpoint, for the type of display considered in this article, we feel that the surface rendering method of the type described here should be preferable to the volume method considered here on the basis of the nature of the information portrayed in the renditions and time and storage requirements.

He goes on further to say:

An issue we have not addressed in this article that is usually considered while comparing surface and volume methods is the structure oriented mensuration and manipulation found useful in interactive surgical planning. These are not possible with the volume methods at present; but with an appropriate data structure (of the type mentioned here) to represent fuzzy structures it is possible to incorporate such capabilities into volume methods. We do now know what it means to visualize, manipulate, and analyze a totally diffuse entity (which has no boundary by any definition), either by its own or in combination with other diffuse or tangible entities.

There are other important motivations to generate a surface from a volume, and many authors feel that a surface is preferable to volume visualization. Analysis or manipulation requires a boundary representation of an object. The current state of isosurface algorithms does not guarantee a valid, Jordan surface under



any conditions. Indeed, the faults are treated in the literature as tolerable errors that can not be avoided in surface rendering. The desire for a robust surface representation from noisy volume motivated the authors development of the SpiderWeb algorithm. I believe that the SpiderWeb algorithm will make surface rendering a valuable modeling tool as well as a visualization technique.

Wallin presented [107] his modification of Marching Cubes in which he disambiguates voxel face cases containing 4 hits with a new face centered vertex assigned a bilinear interpolated value. His algorithms for surface generations consists of two phases. The first is edge generation and cube intersection table lookup. The second phase is edge connection and polygon generation. He keeps hashed lists of signed edges to speed polygon generation. He also uses an *ad hoc* case table approach, citing the number of possible tessellations as

$$(2.1) \quad \frac{s^{n-2}}{(N-1)!} \prod_{k=1}^{n-3} (2k+1)$$

where  $n$  is the number of vertices in the polygon. For  $n = 12$  the number of polygons is 16,796. He cites the need for further refinement in the triangularization of hits which he hopes will come from the mathematical proof of the surface tiling algorithms. I believe that this dissertation is a step in that direction he cites.

## Chapter 3

# The SpiderWeb Algorithm

There is a fundamental ambiguity in two dimensional contouring algorithms [108] [109] that also applies to three dimensional contouring techniques. This ambiguity arises only in the relatively rare circumstance when there are 4 threshold crossings (hits) across a rectangular set of 4 adjacent pixels, or a boxel. This also occurs as 4 hits on any voxel face. This issue seems to have never caused much concern in two dimensional contour lines because there were no applications that measured length or area by contour lines. These issues become important with the use of isosurfaces to make critical judgements in medical imaging of noisy volumes [60] [95], electric impedance tomography, and magnetic source imaging of the brain [110].

### 3.1 Terminology and Definitions

Our development of the SpiderWeb algorithm as a three dimensional contouring algorithm will start with examining the SpiderWeb two dimensional contouring algorithm. This is similar to a number of standard plane contouring algorithms [11] [108]. I show that there are significant ambiguities in two dimensional contour plotting. These ambiguities are more nagging in three dimensions. Figure 3.1 illustrates a simple two dimensional axial MRI slice with contour lines separating brain and not brain. Figure 3.2 is an magnified section of Figure 3.1. The grid is formed by contouring boxes or boxels. A boxel is a box formed by 4 pixels at the vertex of the box corners, and consists of 4 edges along which the isothreshold crossing is calculated by linear

interpolation along boxel edges. A boxel is also a two dimensional analog to the voxel. There are a few instances of 4 contour line intersections on the lattice boxes which cause small islands or bubbles in the contour rings. Note that the isovalue that best outlines the brain also contours about the skull and other bordered, non-brain internal structures. The strength of the SpiderWeb algorithm is that it tracks connected surfaces, and will not jump to a disconnected surface, even if it is at the same isovalue. This 2D contour plot shows all of the contour lines at this threshold, connected or disconnected. This 2D plot is used to pick a seed voxel from a slice to start tracking. Ordinarily, the user with the computer mouse would pick a seed boxel that contained a contour line inside it. This boxel represents the bottom face of the three dimensional voxel above it. Because the boxel face contains contour lines, the voxel face must contain isosurface triangles, and we can use the voxel above the boxel as a seed to start tracking our surface in three dimensional space.



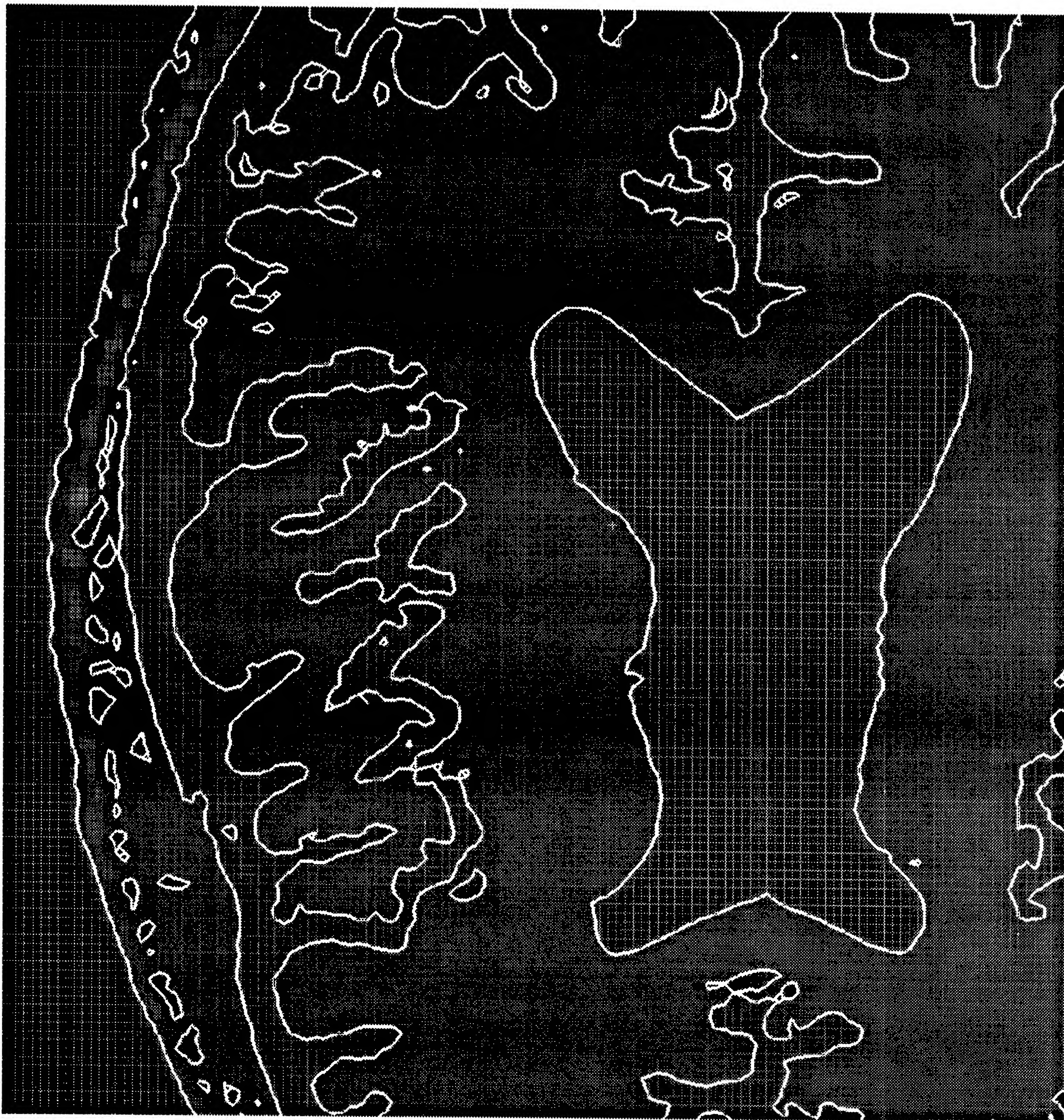


Figure 3.1: A isocontour plot of an MRI axial slice of the brain to preview the SpiderWeb geometry.



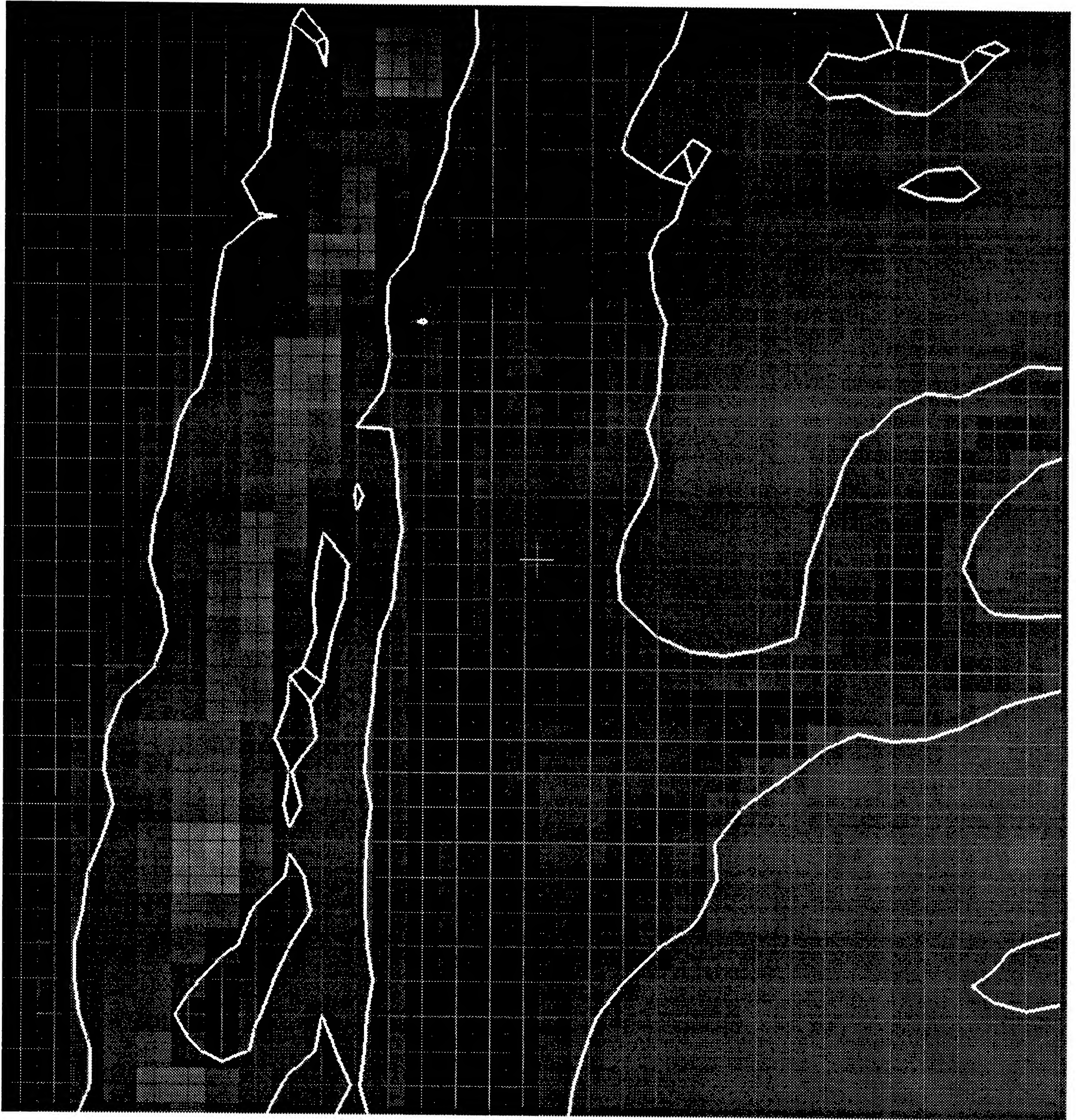


Figure 3.2: A 4X magnification of previous MRI slice showing contour line detail



### 3.1.1 SpiderWeb For Two Dimensional Contour Line Algorithm

Our overall goal is to build three dimensional surfaces in volume data. However, as we will show, this operation is directly analogous to the simpler problem of constructing contour lines in a plane. We will develop the ideas that make the SpiderWeb algorithm unique first in two dimensions, in this section, then show that many (but not all) of these ideas generalize in three dimensions in Section 3.2.1.

The Snyder contour line plotting algorithm, published in the 1978 [11], was used by Elliott *et al.* [109] in a contour line extraction and segmentation system. Elliott states:

This algorithm has been very helpful for the definition of the outline contour of the patient and for segmentation of bony structures or lung tissue in CT images. However, this algorithm completely fails for more complex objects with only small grey-level differences to the surrounding tissue.

The limitations of 2D contouring were cited by Elliott. It is curious that the ambiguities that motivated my research into the SpiderWeb 3D contouring algorithm have existed in 2D contouring literature for quite some time.

#### 3.1.1.1 Mathematical properties of functions and terminology

We define some constraints on the properties of the function that generates density values in a plane. In Section 3.2.1 we continue the development of these properties for three dimensional volume.

**Smoothness.** As we look closer and closer at the underlying function by increasing the density of our sampling, the variability between adjacent samples decreases. More exactly, as we subdivide and sample the boxels more and more finely, the number of 4 hits on boxel edges will tend to decrease to 0. This means that as the resolution of the measurement increases, the variation in the values at higher resolution will tend to decrease as the fineness of the sampling increases. This would exclude self-similar, or fractal functions, which appear infinitely intricate at any observation resolution.

**Continuity.** The function does not have any singularities, breaks, or holes into which the interpolant function can fail. Because we use linear interpolation, we could not see point singularities in a sampling of a continuous data field, even if they were there. It is not possible to observe a point singularity in a volume density by any imaging instrumentality. All of our measurements are the average value of some function accumulated over the smallest resolvable volume of the instrument. Because the density function in our volume is continuous, the isosurface in that density is also continuous. For a function such as mass density, this is always true.

**Boundedness.** From any arbitrary point on the plane, if you travel in a straight line a distance approaching infinity, the value of the density will decrease to zero. The density function has a lower bound of zero, and a measurable upper bound. The area we wish to study must be bounded by zero value in the plane. A hyperbolic or parabolic curve is not possible in a bounded plane.

We now define some common terminology to discuss two dimensional contouring. These definitions will directly generalize for three dimensional isosurfaces. A **pixel** is a point sample of a data lattice. A **vertex** is the corner of a polygon. We refer to a **boxel vertex** as the corner of a boxel. A **line vertex** is one end point of a line segment. An **edge** is a non-interior path between two vertices. A **edge** refers to the line segment between two boxel vertices, and also refers to the line segments between two line segment vertices. The density field we wish to contour is sampled on a data lattice at integer lattice locations  $i, j$  which represent real world locations  $X, Y$ . We make a two dimensional rectangular box with pixels at the vertices of the box, called a **boxel**. A **boxel face** is the plane of the boxel edges and boxel vertices. A **voxel** is the three dimensional extension of a boxel, and a detailed definition is in Section 3.2.2. Basically, a voxel is a rectanguloid box with pixels at each of the 8 vertices. A voxel face is essentially the same as the boxel face. A vertex is **inside** if its value is greater than threshold  $\tau$ , and a vertex is **outside** if its value is less than or equal to  $\tau$ . Along each edge of the boxel, we linearly interpolate density values along boxel edges. By the mean value theorem of calculus, and by continuity of the linear interpolant function, any edge  $E$  that has one vertex  $V_0$  inside or outside and the other vertex  $V_1$  outside or inside, respectively must have some location where the threshold value  $\tau$  hits that edge. The symbol  $\&\&$  represents the logical **AND** condition of the two value comparison tests  $\geq$  and  $<$ . This is defined formally below:

$$(3.1) \quad (V_0 > \tau) \ \&\& \ (V_1 > \tau)$$

There is no hit on  $E$ .

$$(3.2) \quad (V_0 < \tau) \ \&\& \ (V_1 < \tau)$$

There is no hit on  $E$ .

$$(3.3) \quad (V_0 < \tau) \ \&\& \ (V_1 \geq \tau)$$

$$(3.4) \quad (V_0 \geq \tau) \ \&\& \ (V_1 < \tau)$$

There is a hit on  $VE$ . The position  $E_{hx}, E_{hy}$  of hit  $E_h$  along edge  $E$  is given by:

$$(3.5) \quad E_{hx} = V_{x0} + (V_{x1} - V_{x0}) \frac{\tau - V_0}{V_1 - V_0}$$

$$(3.6) \quad E_{hy} = V_{y0} + (V_{y1} - V_{y0}) \frac{\tau - V_0}{V_1 - V_0}$$

A hit occurs when the edge vertex values “straddle” the value of  $\tau$ .

For example, in Figure 3.3, the threshold is  $\tau = 60.0$ , and Vertex  $V_{i,j}$  of edge  $E$  has a value of 90.0, located at  $X = 0$ . At the other end of edge  $E$ , Vertex  $V_{i+1,j}$  has a value of 30.0, and a position of 1.0. The interpolated position of the hit is  $1/2$  of the way between the ends of the voxel edge, at  $x = 50.0$ . Because the edges can be arbitrarily oriented with respect to the world space coordinate system, the  $y$  values of the hit positions also needs to be interpolated if the edge is not oriented along the coordinate grid.

### 3.1.1.2 Only one hit is possible along an edge

For linear interpolation, or any monotonic linear interpolant function we use along an edge, there can be zero or one hit along an edge. While it is possible that a higher order interpolant functions can have as many hits as their order, we have no *a priori* basis to do this. Any function of the type we have discussed above (smooth, continuous, bounded) can be piece-wise represented by a sufficiently small interval linear function. This is especially true in the situation with MRI or CT where the underlying nature of the data is not known. Our only recourse is to get sufficient sampling points, or do the best we can with the data at hand. By taking sufficiently fine samples, we can represent any function as a combination of linear interpolant functions.



**Isosurface Threshold=60.0**

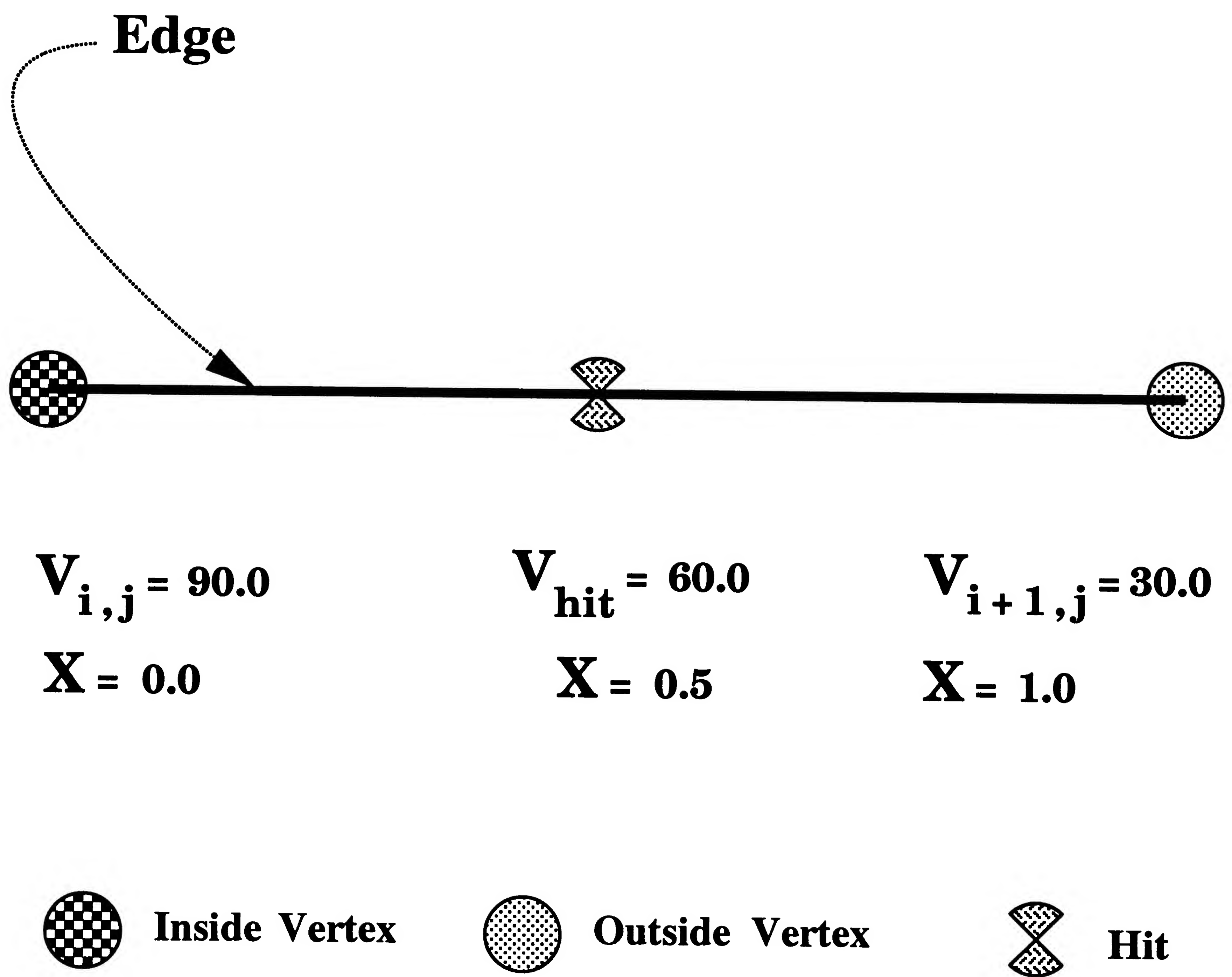


Figure 3.3: Linear interpolation of hit position on edge.



### 3.1.1.3 One hit on a face is not possible

For the face of a geometric figure, such as a boxel or voxel, it is not possible to have only one hit on a face. For any closed path along face edges, **hits must occur in pairs**. The proof by contradiction of this proposition is simple. Figure 3.4 shows the contradiction caused by only one hit. A vertex can have one of two states with respect to a threshold  $\tau$ , and that state is inside or outside. Inside represents a vertex value equal to or above the threshold ( $\geq \tau$ ), and an outside vertex is below the threshold ( $< \tau$ ). A hit represents a transition in vertex state along an edge from inside to outside, or outside to inside. A closed path is a sequence of edges that returns to the starting vertex. The arrow curling around inside the boxel represents a consistent circulation orientation to walk along the edges of the boxel. Each vertex is shared by exactly two edges. We will always return to our starting vertex if we traverse the edges in a consistent order. If we cycle around the closed path of edges in a face when we return to our starting vertex, we need another hit in order to return to our starting state. Without an additional hit, we would return to our starting vertex in the opposite state (inside or outside) from which we left. This contradiction shows that 1 hit is not possible on a face.

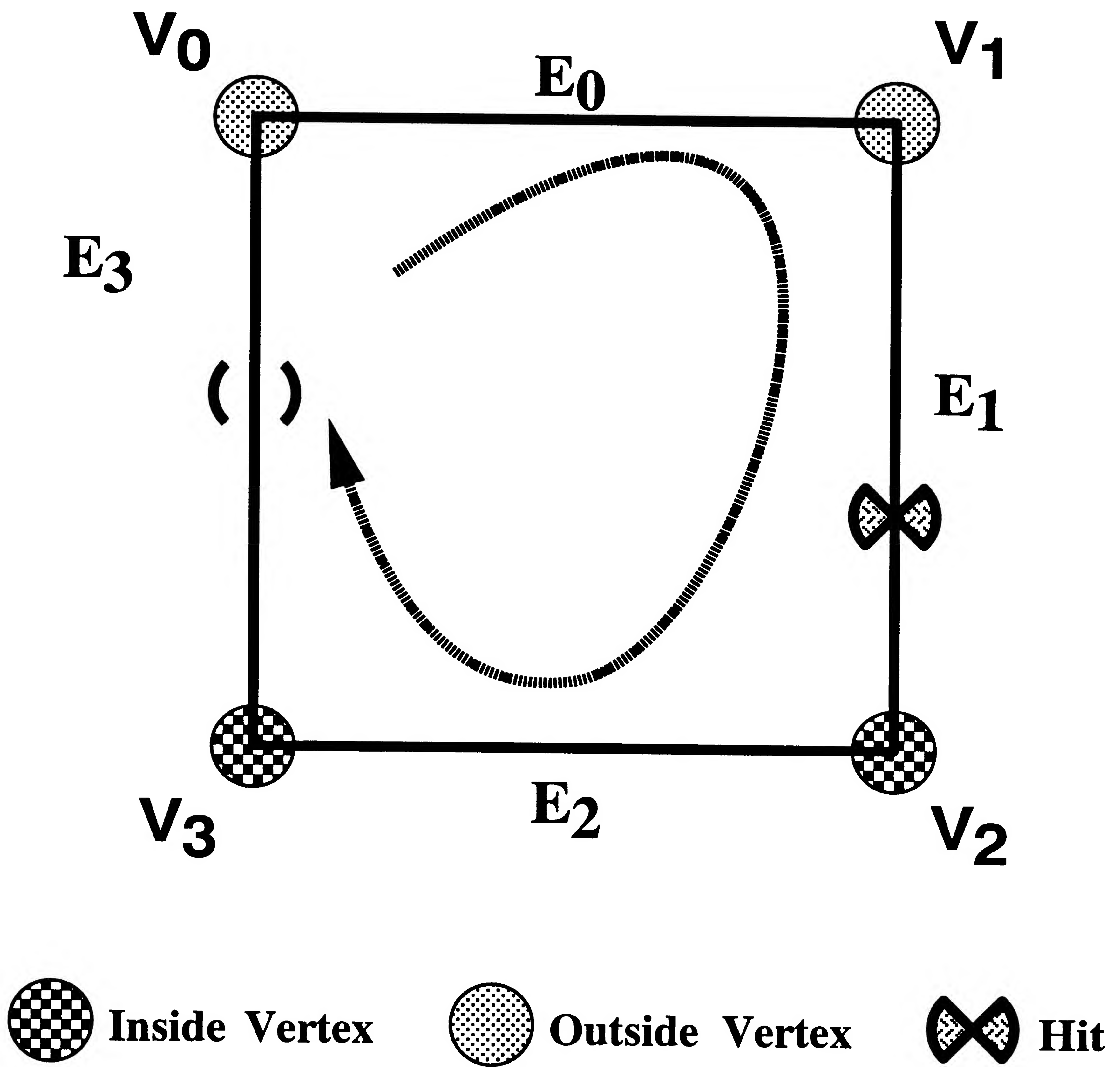


Figure 3.4: Impossibility of 1 hit on a face.

#### 3.1.1.4 Two hits on a face yield unambiguous geometry

For a boxel with only 2 hits, a contour line segment is constructed by connecting each hit with a straight line segment. A number of such segments make contour segments, and these lines make the traditional contour lines and rings.

An important property of boxels (or voxels, for that matter), with only two hits is that there is no ambiguity in the construction of the geometry. We have no doubts about a figure constructed from elements that have only two hits in all of its' edges.

#### 3.1.1.5 Three hits on a face is impossible

We now show that it is impossible to have three hits on a boxel, by proving that only three hits create a logical contradiction. A hit can only occur on an edge that makes a transition from inside to outside, or outside to inside. Referring to Figure 3.5, we cycle clockwise about the edges and vertices of a boxel. Starting at outside vertex  $V_0$ , we travel edge  $E_0$ , to inside vertex  $V_1$ , and  $E_0$  contains a hit. From inside vertex  $V_1$  to outside vertex  $V_2$ , along edge  $E_1$  we obtain the second hit. Continuing from outside vertex  $V_2$  along edge  $E_2$ , if there is to be a third hit, vertex  $V_3$  must be an inside vertex. However, if  $V_3$  is inside, then there will also be another transition along edge  $E_3$ , from  $V_3$  to  $V_0$ , causing a fourth hit. Therefore we can see that in general for any system of edges that forms a closed path, there can only be an even number of hits about the path. **Hits must occur in pairs about any closed path.**

The importance of this observation is threefold. First, it makes the analysis of systems of hits simpler. We only need to consider an even number of hits along a closed system of edges or face. Secondly, any single hit must be completed by another hit somewhere around a closed path. Thirdly, any single hit on an edge that is shared by another boxel will cause an additional hit on some other edge of that boxel. In this sense, this pair of boxels has a connected system of hits. Within a closed path each pair of hits can be connected by the closure requirement. Therefore, we introduce the idea of a **connected voxel face**, where one hit shared by two voxel faces causes the faces to propagate hits. Additionally, we use the term **connected edges** to denote the edges containing and between a pair of hits. Such a pair of hits is said to be **connected hits**. We will show later how we can define a connected system of hits by following connected faces and their connected edge hits.



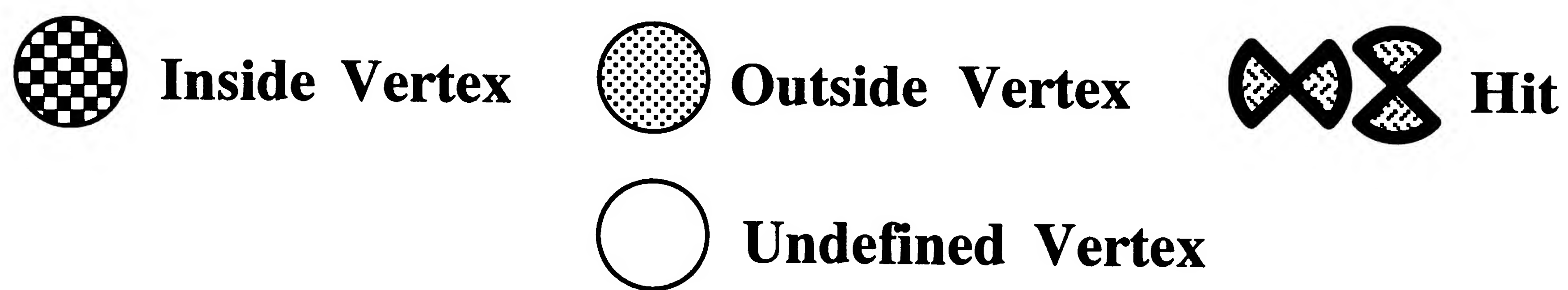
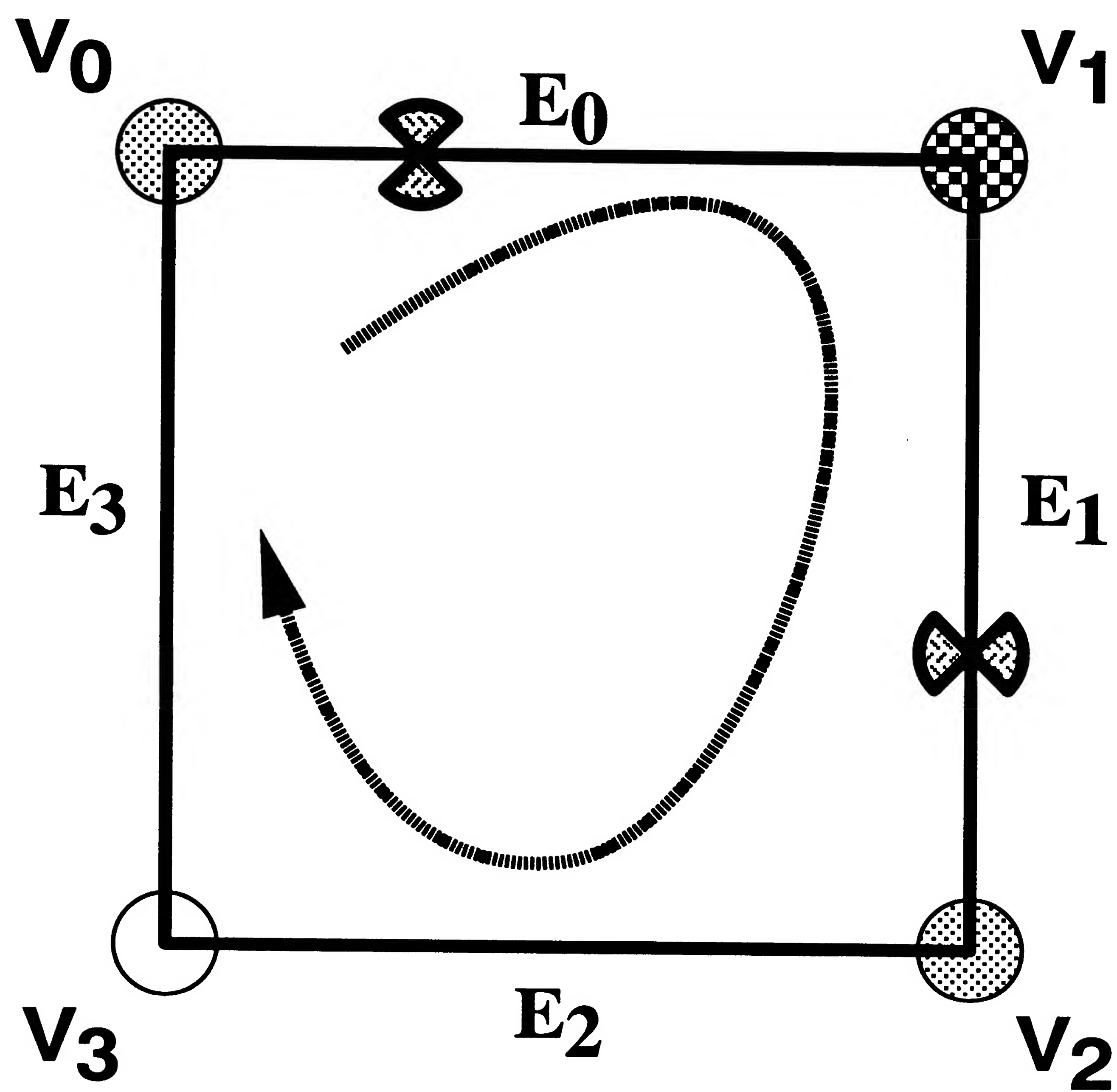


Figure 3.5: Impossibility of 3 hits on a face.

Figure 3.6 shows an example of an impossible surface configuration. Two rules are violated. There are two hits along a voxel edge. There are only three edges containing hits. We have shown in Section 3.1.1.2 that an edge can have only one hit. We have also shown in Section 3.1.1.4 that a face must have 2 or 4 hits. Therefore, this proposed surface configuration can never occur.

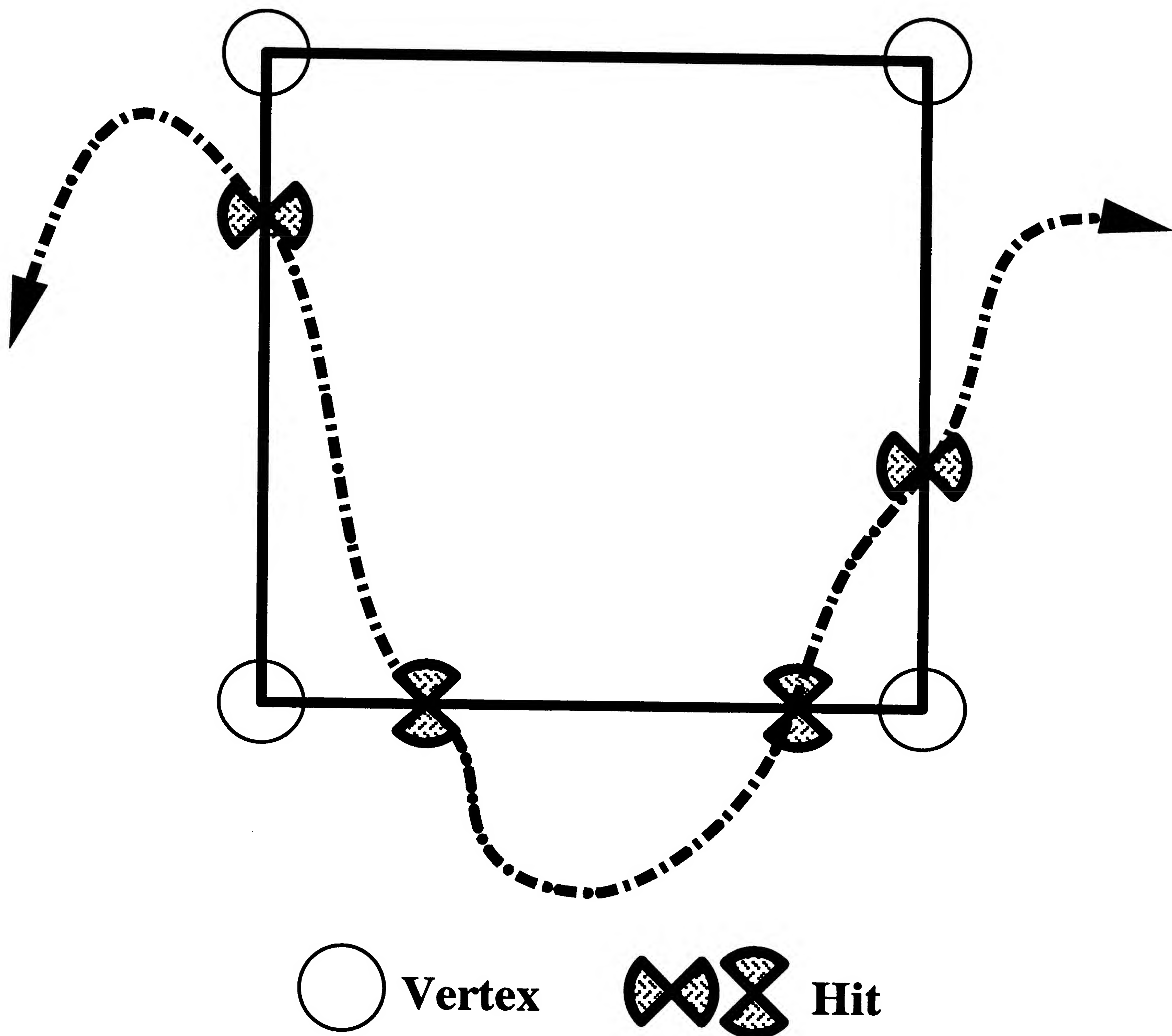


Figure 3.6: Impossible surface configurations.

### 3.1.1.6 Four Hits on a Boxel Face

The situation with 4 hits on a boxel face creates four possible choices ways to knit these four hits into contour lines segments. We will now examine these alternatives, and explain why this situation is important.

While these situations are rare in clean data with strong gradients, we will show that the topology of the surface is critically defined at faces with 4 hits. When two surfaces pass closely to each other, there is a possibility that at some nearby threshold value that the surfaces will kiss (osculate), and at a slightly greater threshold value, the surfaces will join(c.f Figure 3.20). These regions where this joining or separation of surfaces is quite important in defining the topology of the surface for a range of thresholds. Because a surface can never join or separate in a face with two hits, the topology can only change in a face with four hits. Therefore, these situations, while not as common as two voxel hits, critically define the global nature of the contour lines or surface we are building.

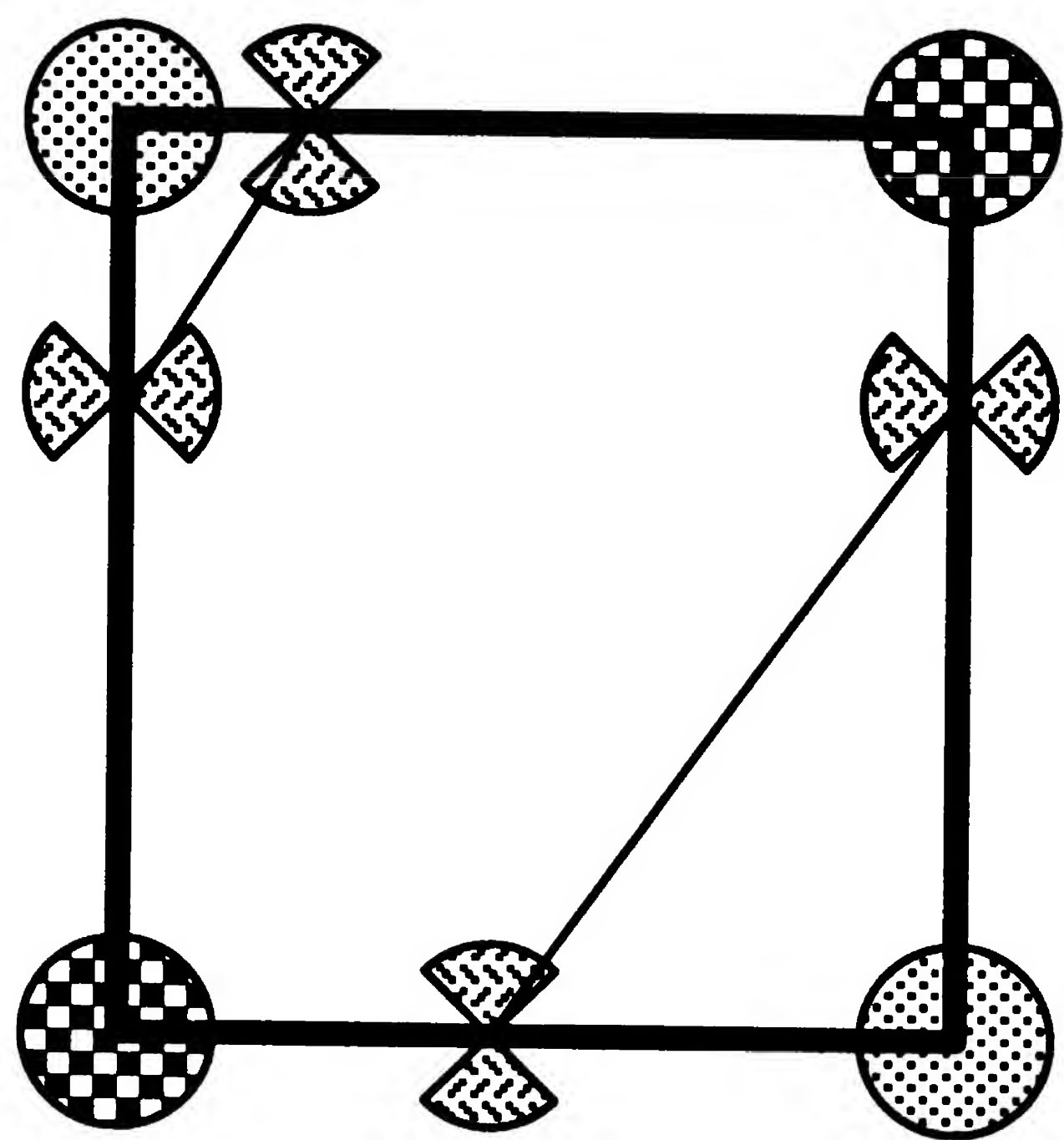
In Figure 3.7, Example boxel C, showing two lines intersecting, would not make non-intersecting contour lines, and is not a viable choice. More exactly, Example boxel C would not produce a Jordan curve, which is a curve that segments inside from outside regions of the plane.

Example A and B seem to represent reasonable choices, and we need a basis to choose A **OR** B. We will examine various techniques to make a rational choice in choosing which way to knit hits. We will also show that there are a number of situations where there is no way to decide which way to knit. These singular situations define the topology of the surface in the vicinity of the threshold value.

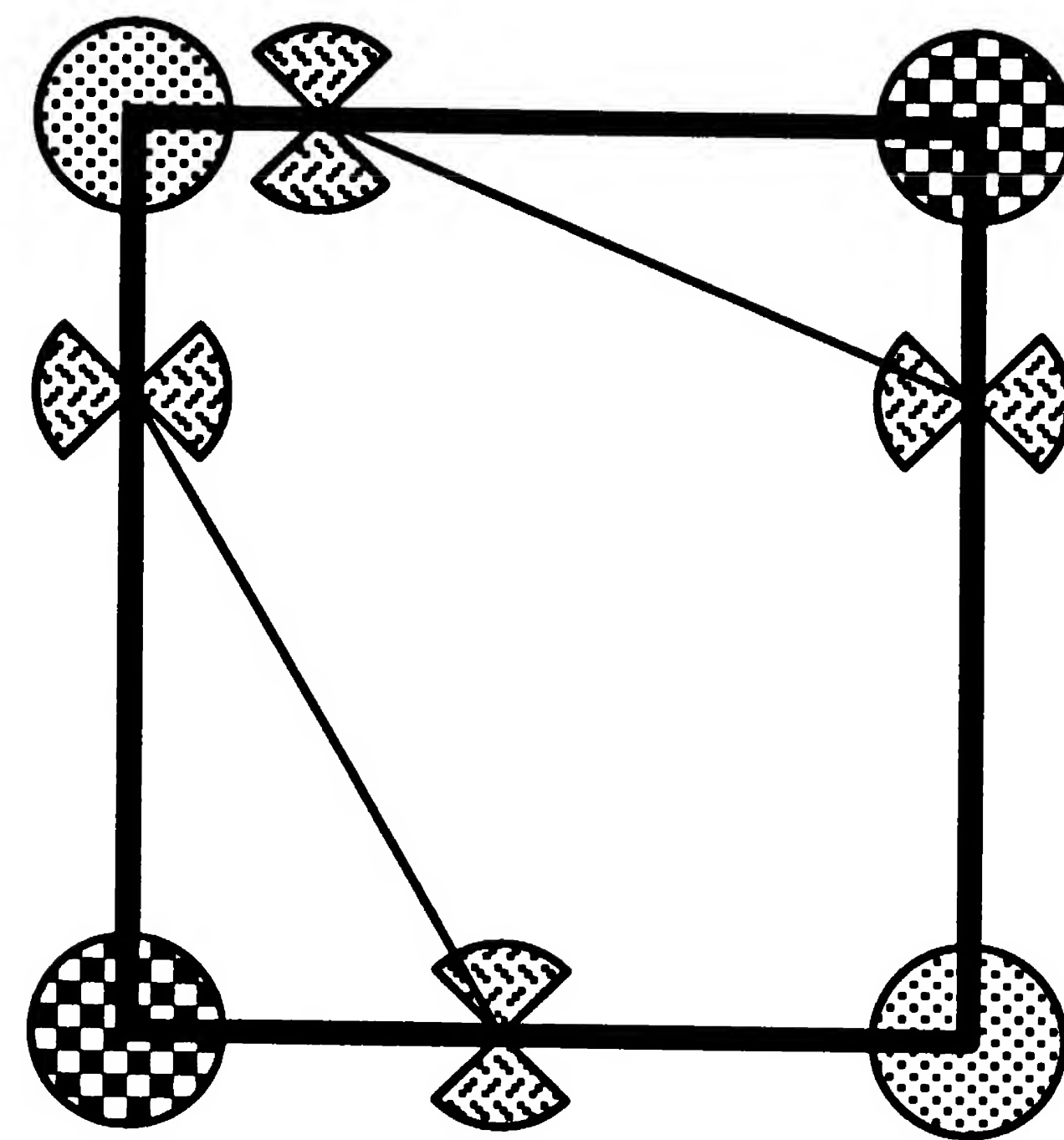
We can choose A **AND** B, which is shown in Figure 3.7 Example D. This choice represents an expression of the ambiguity with a two dimensional **bubble**. In three dimensions, we show that this configuration of 4 hits on a voxel face also produces a bubble in the isosurface.

The standard technique for dealing with this ambiguity has been to guess consistently. This can mean always knitting as in example A or B. The SpiderWeb algorithm approach is to knit all possible exterior triangles into a bubble as an expression of this fundamental ambiguity. This ambiguity arises rarely in clean data. However, the 4-hit situations arises in the presence of noise, excessive curvature of the surface with respect to the edge length, and with weak gradients.

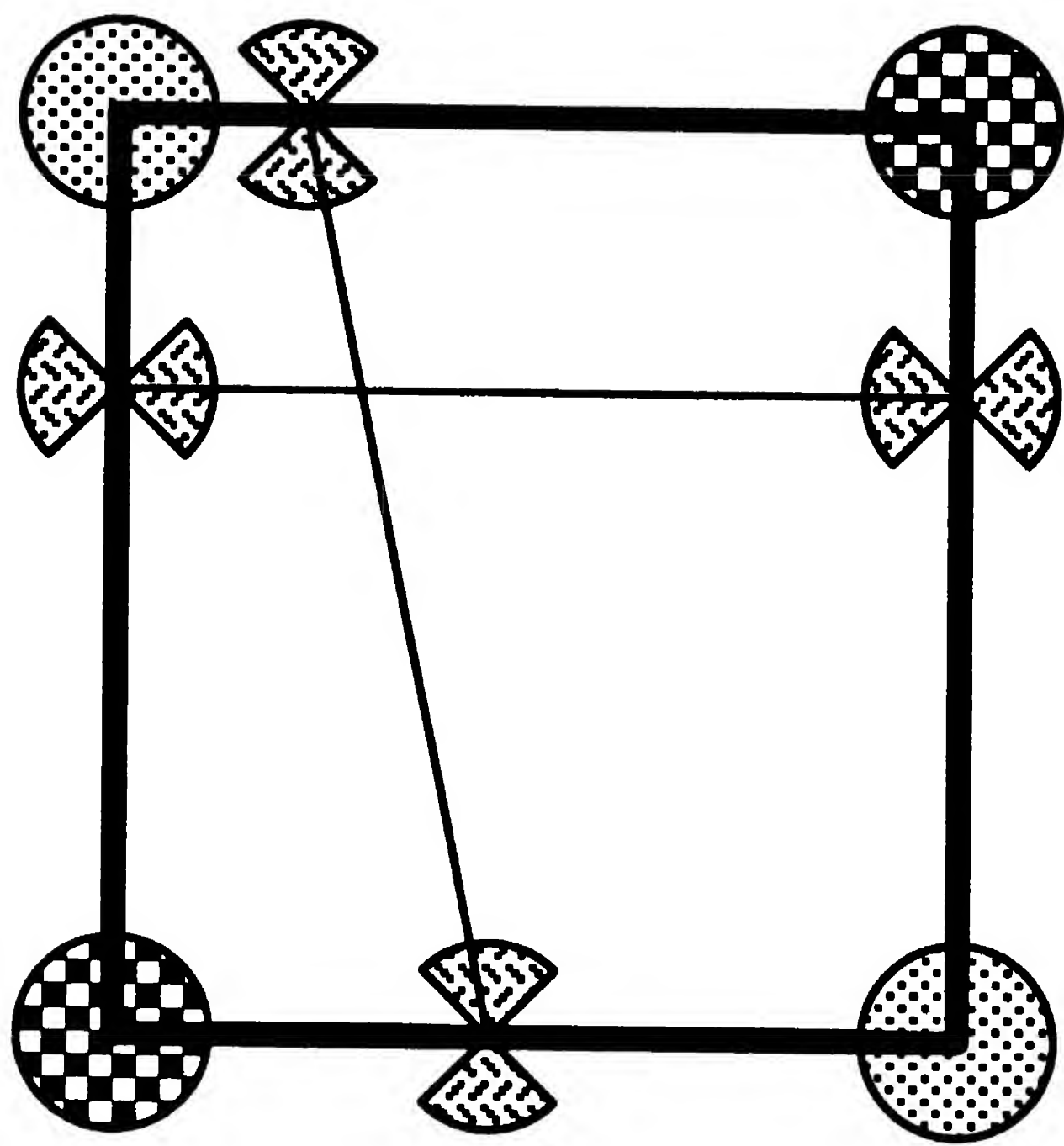




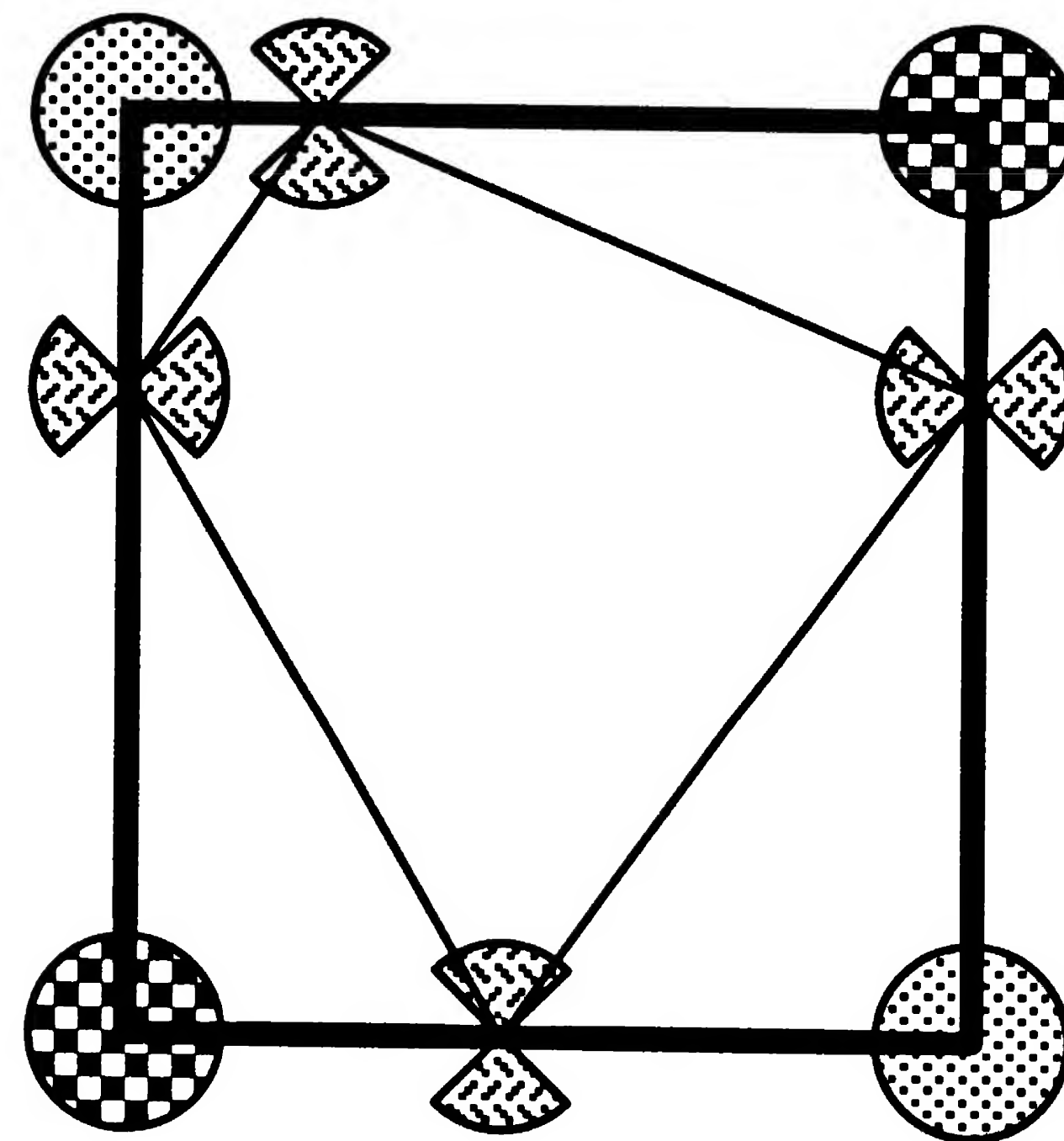
**A**



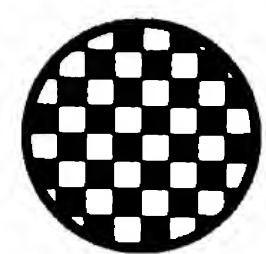
**B**



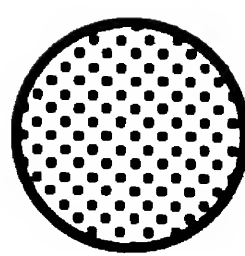
**C**



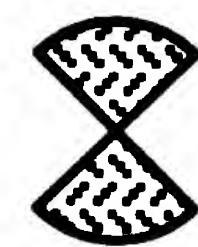
**D**



**Inside Pixel**



**Outside Pixel**



**Hit**

Figure 3.7: Ambiguity in 2D Contour Lines.

### **3.1.1.7 Connected Boxel Faces**

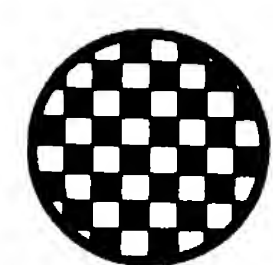
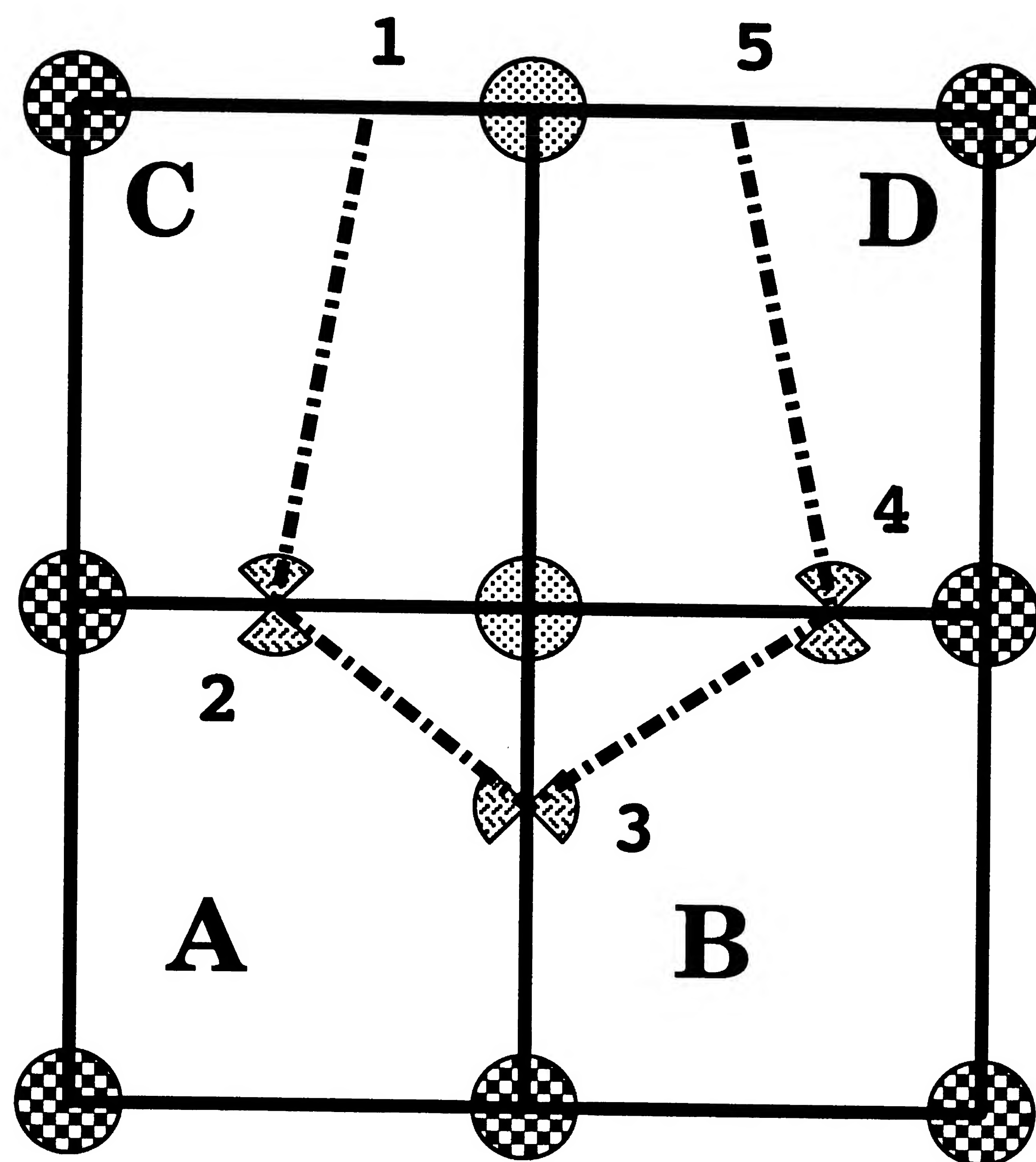
A pair of faces are connected if they share a common edge that has a hit on it. They are connected in the sense that the presence of a hit requires at least one more hit on that face, and that means that a contour line crosses that face. If one is tracking a contour line, and you enter a face, you know that you will exit that face on another edge. In a sense, one hit causes the next by the fact that hits can only occur in pairs. Figure 3.8 shows 4 boxels, A, B, C, D. Boxel C and A share Hit 2. Therefore C is connected to A. However, Boxels C and D do not share a common edge with a hit, and they are not connected. The significance of this concept is more apparent in three dimensional voxels than in two dimensional boxels.

### **3.1.1.8 Connected Edges**

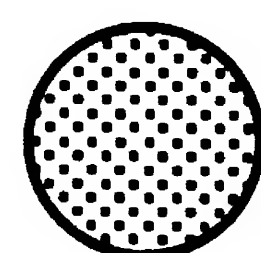
As long as there are only two hits on a face, there is no ambiguity in how to knit contour lines. We have defined connectivity in terms of faces only. Now, we consider connectivity in terms of edges within a face. As long as there are only two hits on a face, there is no ambiguity as to how to knit contour lines from hit points. Figure 3.9 shows four Boxels labeled A, B, C, D. The only essential difference between Figure 3.8 and Figure 3.9 is that we have flipped the lower left hand corner boxel vertex from inside to outside. This causes 4 hits on boxel A, numbered 1, 2, 3, 4. The question is how should these hits be knit? This question is answered two ways with SpiderWeb A and SpiderWeb B algorithms. While we are considering only 2 dimensional contour lines, the following arguments apply in three dimensional surface construction. The three dimensional analysis uses information from the third dimension that is not available in the two dimensional situation.

### **3.1.1.9 SpiderWeb A Hedges**

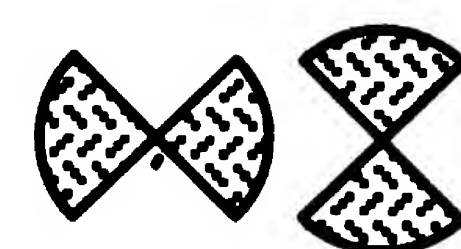
The SpiderWeb A solution is to make a bubble. Figure 3.9 shows a parallelogram of lines that form a 2 dimensional bubble. More exactly, we knit all possible exterior paths between the hits, as is also shown in case D of Figure 3.7. This is the most conservative approach, and produces contour plots such as are shown in Figure 3.2. A number of contour line bubbles are visible in the figure as multiply connected line segments.



**Inside Pixel**



**Outside Pixel**



**Hit**

Figure 3.8: Faces are connected when they share an edge hit.



### 3.1.1.10 SpiderWeb B Chooses

How can we decide which way to knit the lines? The conventional wisdom is to consistently guess. We would like to separate the two system of lines that seem to be converging on our boxel face with 4 hits. How can we separate them ? One choice will generate Topology A in Figure 3.9 and the other choice will generate Topology B. We examine two methods to disambiguate our choice. The first is to make a consistent choice, and the second is to interpolate an additional vertex by triangular decomposition.

We will use our boundedness assumption about our data from Section 3.1.1.1: High values are inside, and are always surrounded by low values. Far away the density is zero, and so we make the choice to surround high values.

### 3.1.1.11 The SpiderWeb B Choice

The SpiderWeb B choice is to surround the high, inside vertex, or the low, outside vertex. We choose to consistently surround high, inside vertex.

Let us track our curve growing through boxel C of Figure 3.9. We emerge into boxel A at hit  $H_1$ . We can scan to the high, inside vertex, or the low, outside vertex to find the next hit. Our consistent choice, which depends on the nature of the data, is to “surround high, inside”. Therefore, we to toward  $H_2$ . We have found the hit to pair with  $H_1$ , and we exit, leaving  $H_3$  and  $H_4$  unexamined. We have built line 2, we do not build lines 1 and 3, and consider line 4 to be a separate, unexamined line. We can come around and build line 4 separately by reentering the boxel A containing line 2. The important idea to consider is that we will enclose the high region to the left, and separate it from the high region on the right. All of our contour lines will no longer be multiply joined, and they will all be Jordan curves. The problem with this approach is that we are making connectivity decision based on distances potentially much smaller than the edge length, or the resolution of the instrument. The significance of these artificial separations has yet to be explored, and this may be artifactual.

We could also choose to enclose low values, by knitting hits  $H_1$  and  $H_3$ . We would be choosing to “surround low, outside”. This would effectively separate the two low vertices diagonally through the voxel. This would yield a different topology, and there is no basis in two dimensions to choose this approach over the “surround

high, inside” rule. We can use the SpiderWeb A approach because it makes less assumptions in its representations of ambiguous contours. We need more information to eliminate the 4 hit boxels. Indeed, we know we have enough resolution when all boxels containing 4 hits are eliminated by getting more, and smaller boxels where we have 4 hit boxels, if we could adaptively get more data. When we do not know, SpiderWeb A does not assume. Ambiguity in the data generates bubbles. SpiderWeb B will make an assumption that may be wrong.

Figure 3.9 shows essentially two different topologies. If we knit high, we will connect the two outside regions via path labeled Topology B. If we knit low, we connect the two high regions labeled Topology A. We can see that the orientation of our connection depends on our underlying assumptions. Alternatively, we can not make a wrong assumption, and build a bubble at boxel A. If vertex 1,1 is equal to vertex 2,2 there is no gradient along the Topology B path. Similarly if vertex 1,2 equals vertex 2,1, there is no gradient along Topology A. Under these circumstances we must build a bubble, as the topology is undecidable [77]. If there is a slightest gradient across the diagonal vertices, we can use that information to decide which way to knit. However, in a binary segmentation, with vertex values of 1 or 0, there are no possible grey values of gradient to disambiguate the topology. Our global assumption of SpiderWeb B is similar to flooding the data surface with water, and separating regions based on the streamlines of the flowing water. The water would never flow up into high, inside regions. It would only flow down, to outside, low lying regions. This physical intuition of this analogy for real data sets provides a concrete basis to disambiguate medical image data sets. This orientation is set by the boundedness assumption of our data set.

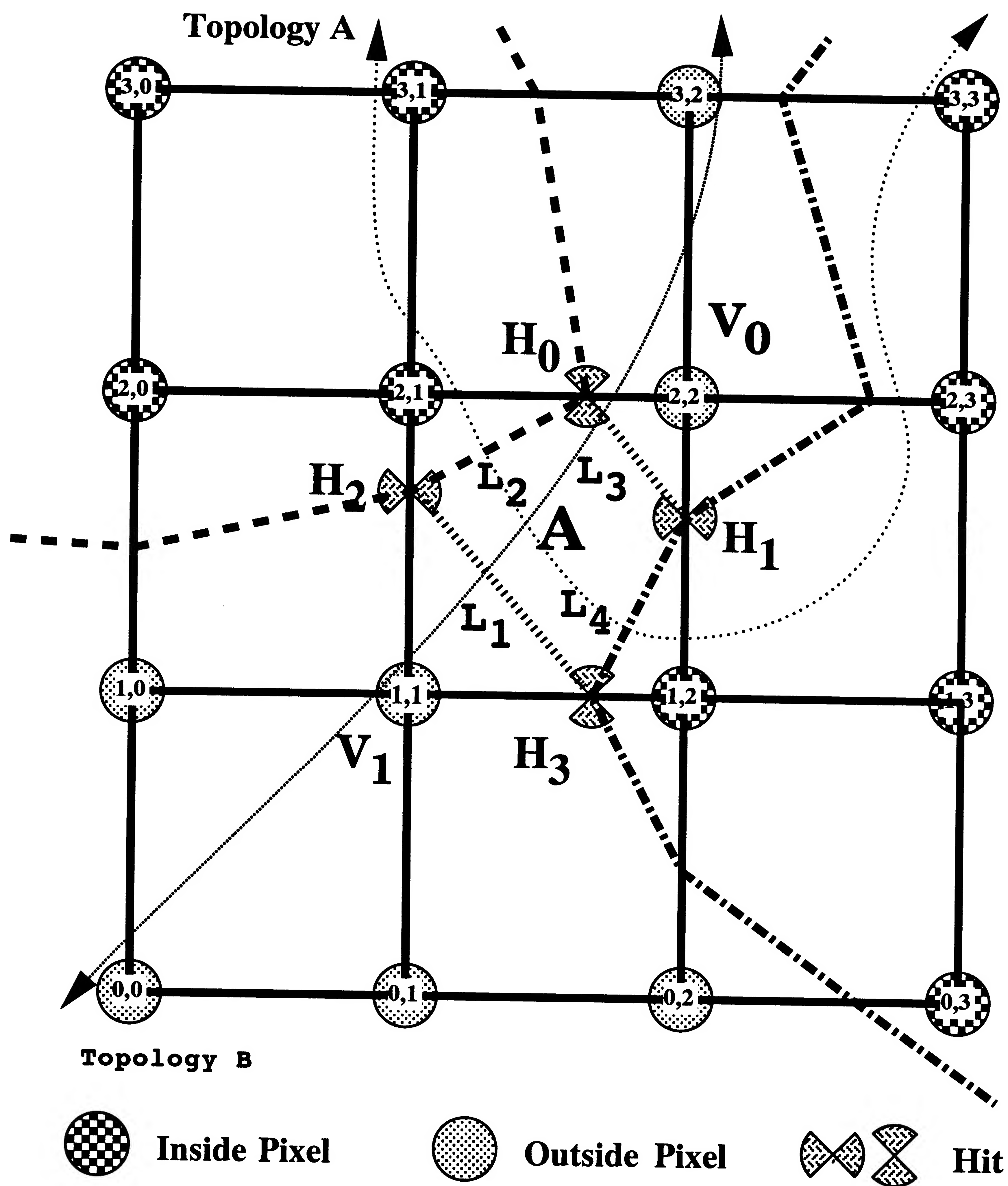


Figure 3.9: Two topologies depending on knitting choices.



### 3.1.1.12 Boxel edge-wise continuity

A contour line must close and form a ring. The above is true if every contour line vertex is shared by two line segments (edges) in a zero bounded plane. Another way to say this is: what goes into one boxel edge from one boxel must emerge from the adjacent boxel opposite (complementary) edge. In this way, a contour line can never stop, have gaps, or otherwise have bad geometry. In two dimensional geometry with contour line segments, this seems to be a trivial condition. It is included here because, as we shall see in Section 3.2.2.2, many three dimensional surface algorithm implementation fail this simple continuity condition for triangle meshes (c.f Figure 1.2).

### 3.1.1.13 Adaptive refinement

We have seen in Section 3.1.1.6 that ambiguous faces have 4 hits on their edges. If we can obtain more data when we are confronted with an ambiguous face, we can possibly replace the 4 hit face with more, smaller faces. By our smoothness assumption in Section 3.1.1.1, as we take finer samples of our data, the variability of the sample values will decrease. We can adaptively reduce the size of the boxel by getting more information at a finer resolution. Alternatively, we can interpolate new vertices when we can not obtain more, new information. We continue refining the gridding until there are only 2 hits on all of the boxel faces(The “No more 4-hit faces” rule). Figure 3.10 shows a scheme for adaptively resampling boxels. Each boxel face that has 4 hits is subdivided until there are only two hits on each face. Does this eliminates ambiguity by getting more sample information ? Can adaptively interpolating the information that is available eliminate ambiguity ?

This appears to be a valid alternative for contouring an implicit function, where we can always calculate finer and finer values. This approach is useful in super-computer numerical simulation experiments. A “No more 4-hit faces” rule provides a well defined endpoint for adaptively gridding complicated models. However, for an MRI or CT instrument, there are resolution limits beyond which we can not obtain any more meaningful information. The danger when we interpolate past the resolution limit is that we are creating interpolation artifacts. Increasing the resolution would only return noise, which would not reduce the number of small boxels with four hits on a face. This would also violate our smoothness criteria in Section 3.1.1.1.

There are certain classes of faces and functions that will never reduce the number of 4 hit faces no matter how finely we resample the data. Consider a binary segmentation, where each vertex can have a value of 1 or 0. If the underlying function always produces 4 hits no matter how you resample it, then adaptive resampling will never resolve the ambiguity. Such a boxel would occur directly over the saddle point of a saddle surface. The saddle point of a saddle surface is the point on the saddle where the curvature inflects, or the sign of the Gaussian curvature changes. Thus, it would appear that we can never eliminate 4-hit faces. However, I believe that these faces hold key information to the surface topology. Our goal should be understanding these singular faces, and the threshold values that produce them. Once we know where they are, and what threshold values produce them, then we can make intelligent threshold value choices to obtain clean, segmented surfaces from our data (c.f. Section 3.2.5).

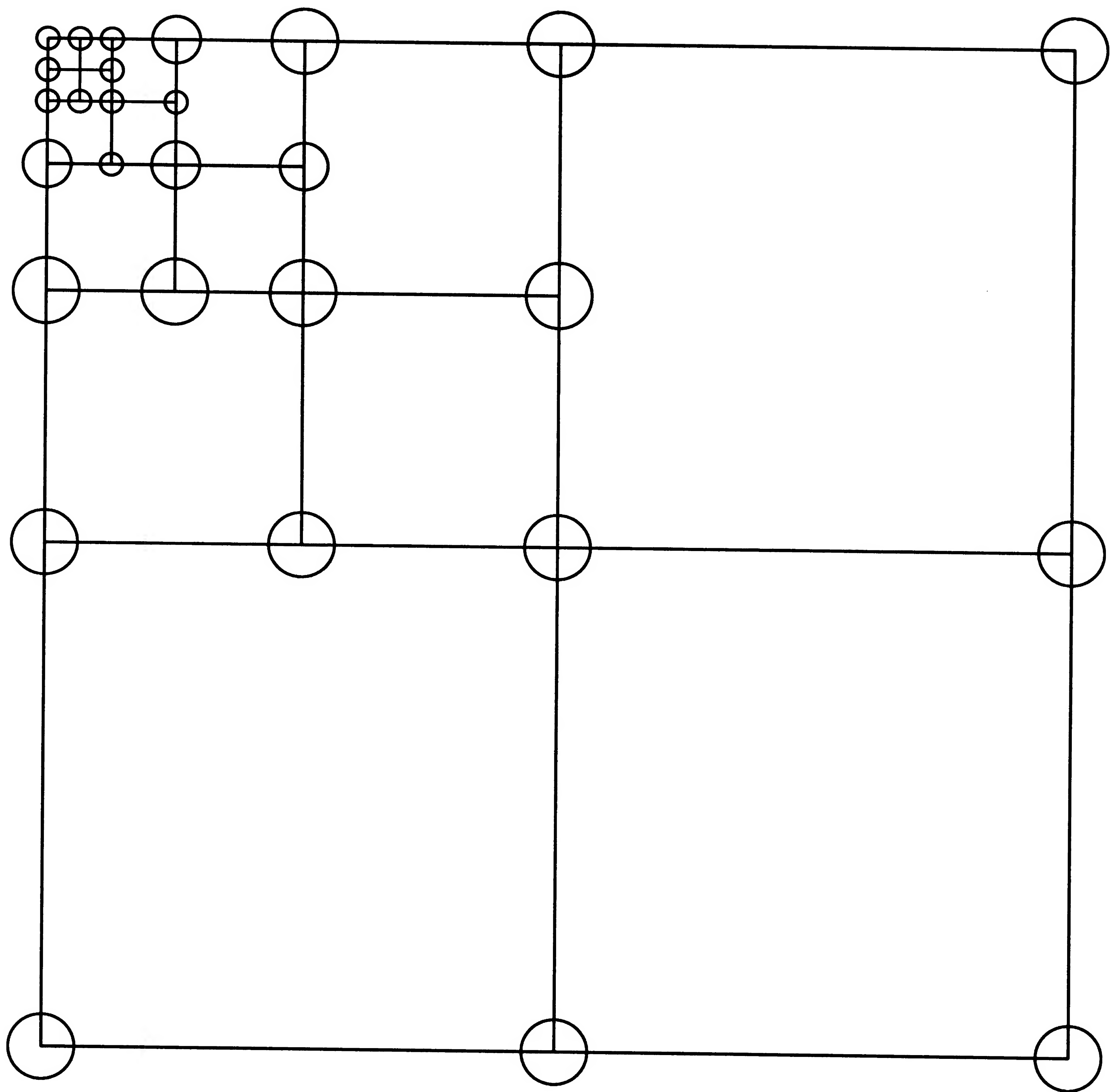


Figure 3.10: Adaptive boxel sampling faces.



### 3.1.1.14 Triangular decomposition

We have in Section 3.1.1.5 that it is impossible to have an odd number of hits on a face. For a three sided figure, it is clearly impossible to have four hits. A possible way to resolve ambiguous faces is to decompose into two possible triangular boxels (simplicial complex). Alternatively, if we can obtain additional, more finely resolved data on triangular samples, we eliminate interpolating new vertices. In three dimensional surface construction, ambiguous cuboid boxels are subdivided into tetrahedral voxels. Does this eliminate any ambiguity?

If our data is collected on a rectangular or square lattice, we can decompose our squares into triangles (triangular decomposition) with one additional edge two ways, as shown as triangularizations  $A_0$  and  $A_1$  in Figure 3.11. With the two different triangularization schemes, we can see from Figures 3.13 and 3.14 that we obtain two different and distinct topologies. We have not created a unique geometry by re-tileing with triangles. Instead, we find that we also have an alternate geometry due to the alternate triangularization. Frey and Gautherie [87] constructed a tetrahedral decomposition of the cuboid with 48 tetrahedra, but did not prove the combination unique.

The remaining triangularization alternative constructs an additional vertex at the center (centroid) of the box, and builds 4 new edges to that vertex. Figure 3.12, is a unique triangular decomposition, but it requires a bilinear interpolated value for the new vertex. It is not been established that this is a correct geometry. We will show in the three dimensional SpiderWeb that this additional point is required for disambiguation in three dimensions, and it can be done without any interpolated vertices.

We can show that a triangularization of the square with only one triangle edge will never disambiguate a configuration of 4 hits on a face. This can be seen in Figure 3.11. The only way for a face to have 4 hits for the diagonal vertices to have the same inside/outside state with respect to any threshold. The two diagonals must have the opposite state, but the vertices of this added diagonal edge can never have the opposite state. Therefore, there will never be a hit on an added diagonal edge to a 4 sided face. Therefore, no new topology will be constructed that will solve our ambiguity. If we have four hits on a face, the diagonal vertices must have the same inside/outside state with respect to some threshold. Creating an edge between vertices with the same inside/outside state will never generate a new hit.

Triangularization by the addition of a vertex and 4 additional edges by bilinear interpolation can disambiguate 4 hits on an edge, but only if the differences from the mean of the 4 face vertices are different from each other. The vertex values across the diagonal must be different. If they are all equal, then the interpolation value of the added vertex will be exactly on the threshold, and will cause a point where all of the contour lines will touch.

This, I believe, is a singularity that represents a change in the topology in the surface sheet with a change in the threshold value. The best way to eliminate the problem is to choose a different threshold value. We need techniques to map topology versus threshold to select thresholds that avoid these singular topologies. However, these singular points may define the topology in important ways. This mapping is discussed in Section 3.2.5.

A possibly important value of the SpiderWeb B algorithm is that it can contour any shape boxel (or voxel) face without decomposing it into triangles.

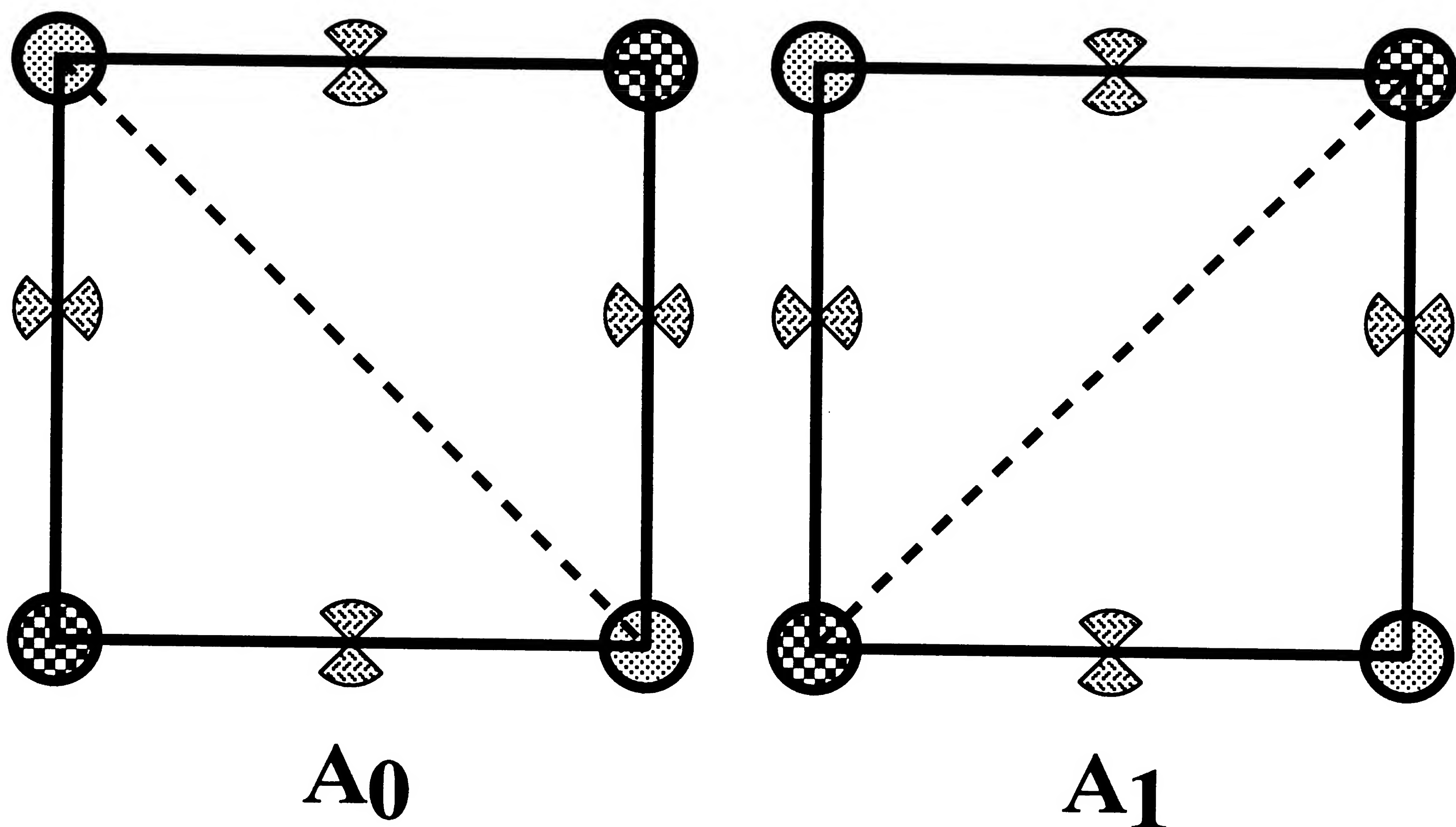


Figure 3.11: Adding an edge to decompose a square into triangles.

Figures 3.15 and 3.16 shows two possible states of the interpolated vertex. Bilinear interpolation of the values for the interpolated vertex  $V$  can result in it having two possible state with respect to a threshold  $\tau$ . If it is inside, we have topology A. If it

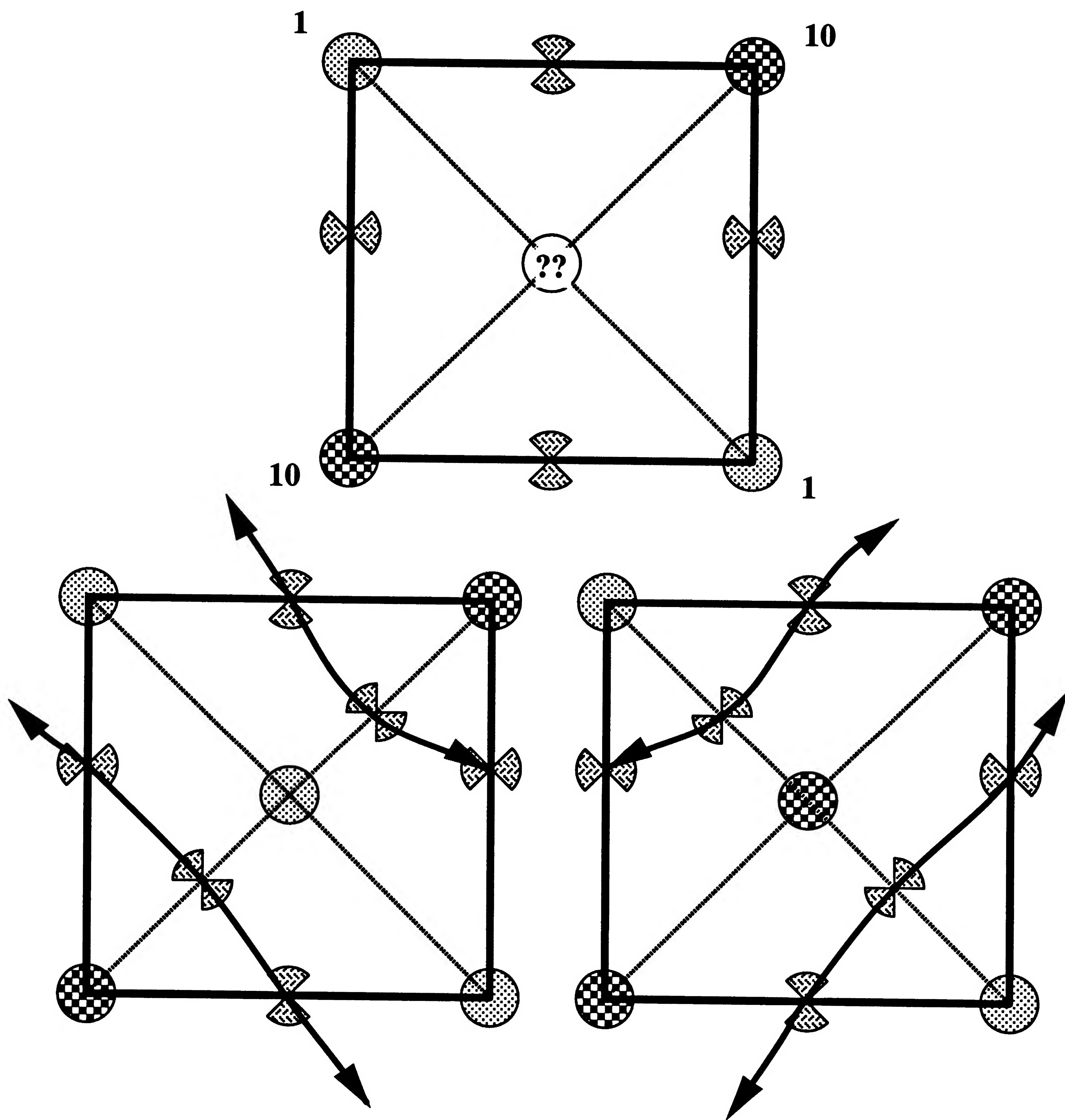


Figure 3.12: Adding a new vertex can disambiguate the 4-hit face.



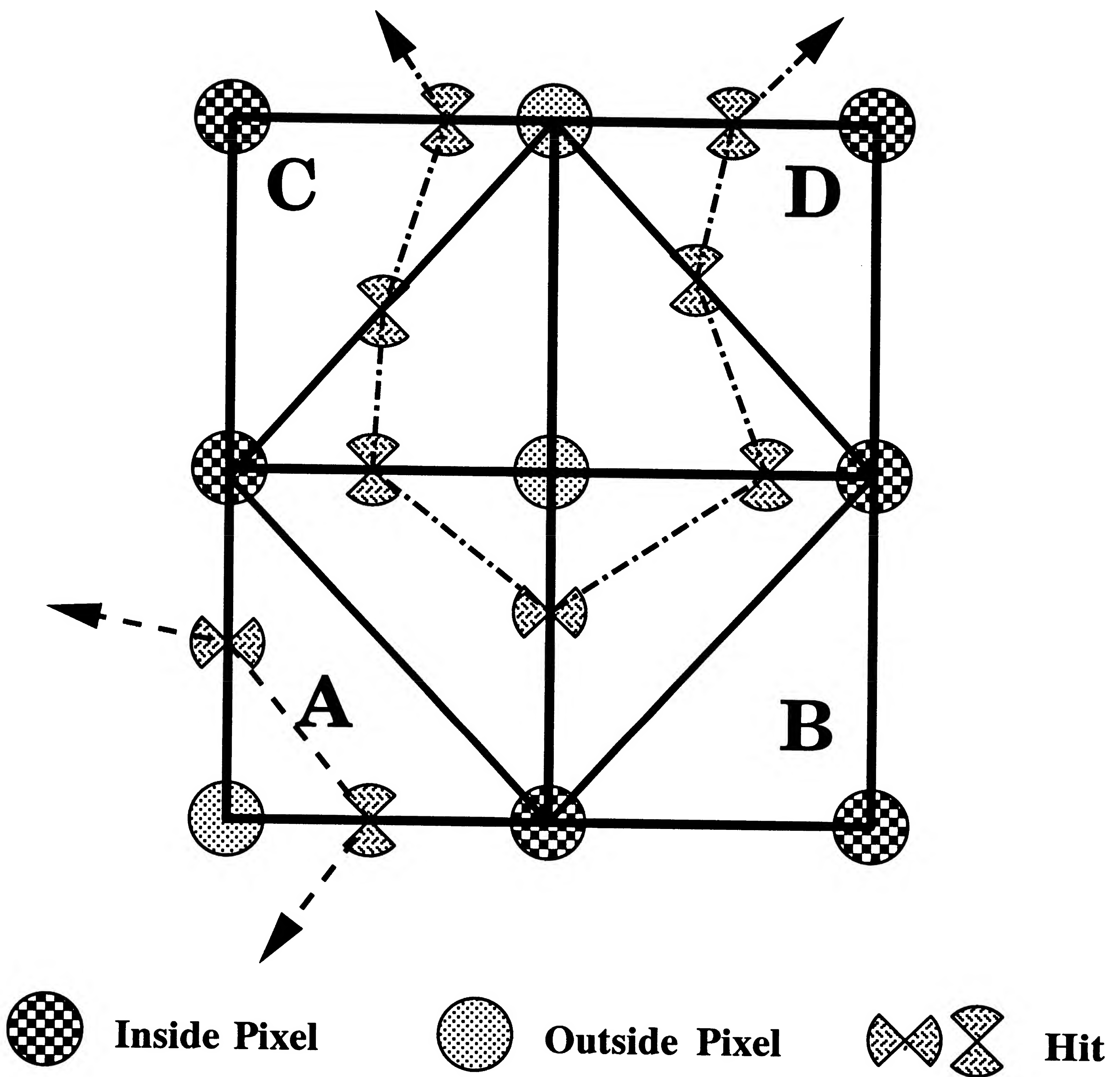
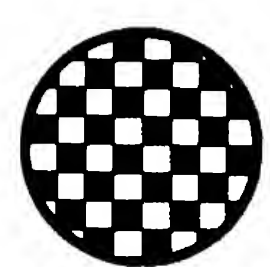
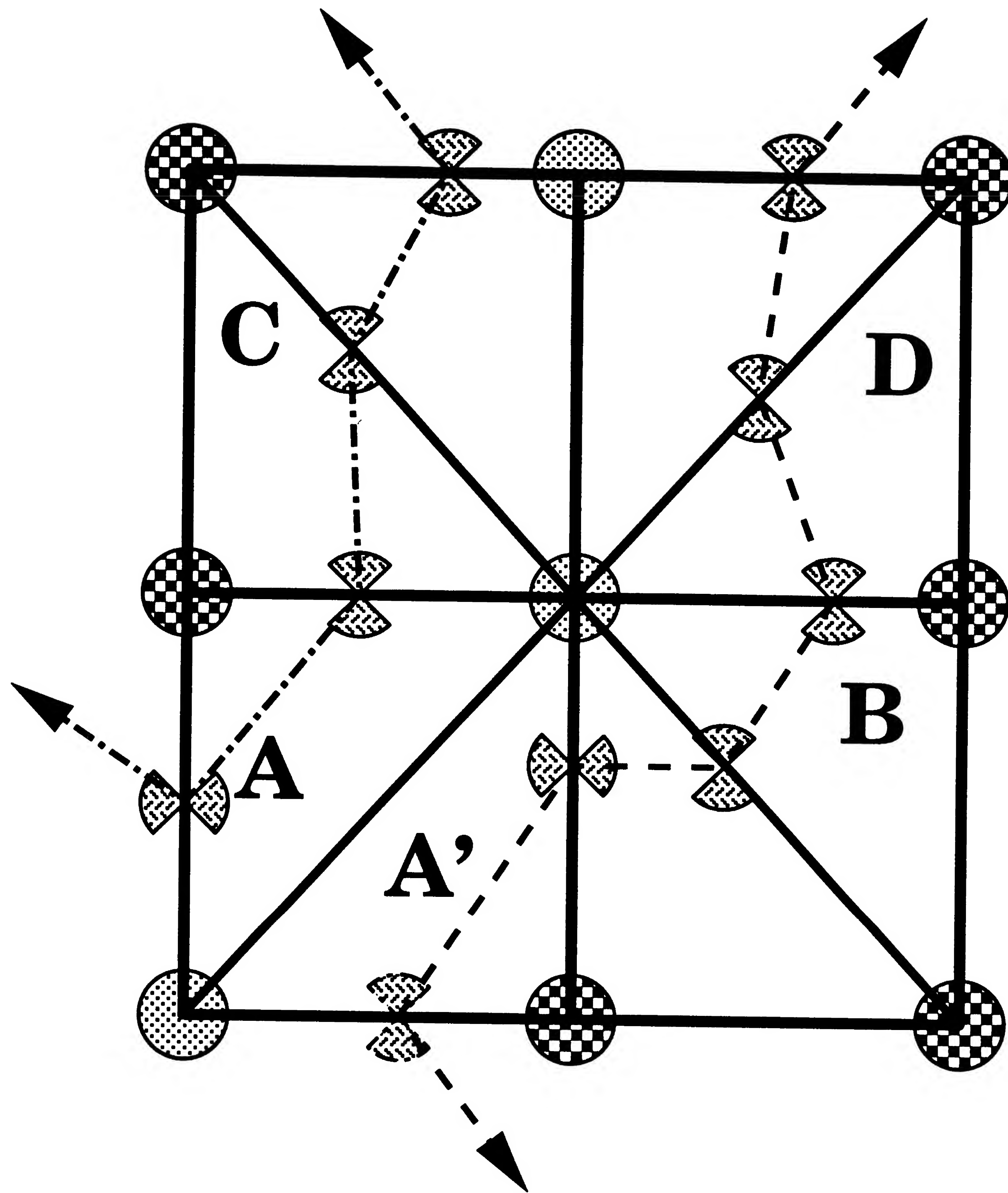
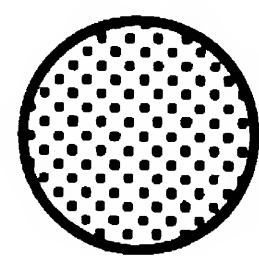


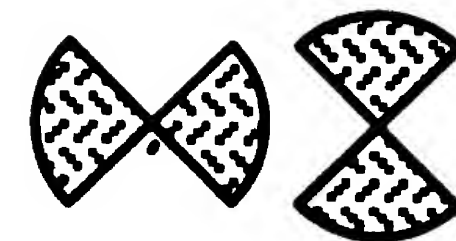
Figure 3.13: Triangular boxels can only have 0 or 2 hits.



**Inside Pixel**



**Outside Pixel**



**Hit**

Figure 3.14: Triangularization of a square boxel is not unique.

outside, we have topology B. This is a possibly worthwhile technique to disambiguate line knitting. However, if  $V_1 = V_0$ , then the gradient along the edge between  $V_0$  to  $V_1$  will be zero. There can be not be a hit if there is no difference along an edge.

The SpiderWeb algorithm chooses to surround high vertices because of the global assumption on the data that high values are bounded. A physical analogy is we make a *bas* relief of the data slice, where the surface height corresponds to the density function value. This is shown in Figure 1.4. Consider rolling a ball bearing down the raised function surface. A bright region would have a higher elevation than a lower, dark region. Our assumption that far away has zero value, is similar to the assumption that far away is the bottom of a potential field. Our ball bearing will head down the gradient until it gets to the outside, or stuck in a blind well. As we scan through the data field, assume that we enter along an edge of a 4 hits boxel. If we have a boxel with only 4 hits on its edges, we can interpolate a bilinear, or quadratic surface across the face of the boxel. The height of the surface is the value of the interpolated density function. We can use our interpolated *bas* relief surface, and roll our ball bearing along the saddle surface from our entry hit point. It will roll downhill toward our exit edge (not exit point). It will never roll uphill over a potential hump. We will use that hump use to separate and disconnect contour rings. A ball bearing will not connect regions that are separated by a ridge of high values. A rigorous proof of the validity of this assumption is in preparation by Karron *et al.* [111]. Our experience with this algorithm has been satisfactory, and no topological contradictions have been detected by the software as yet.



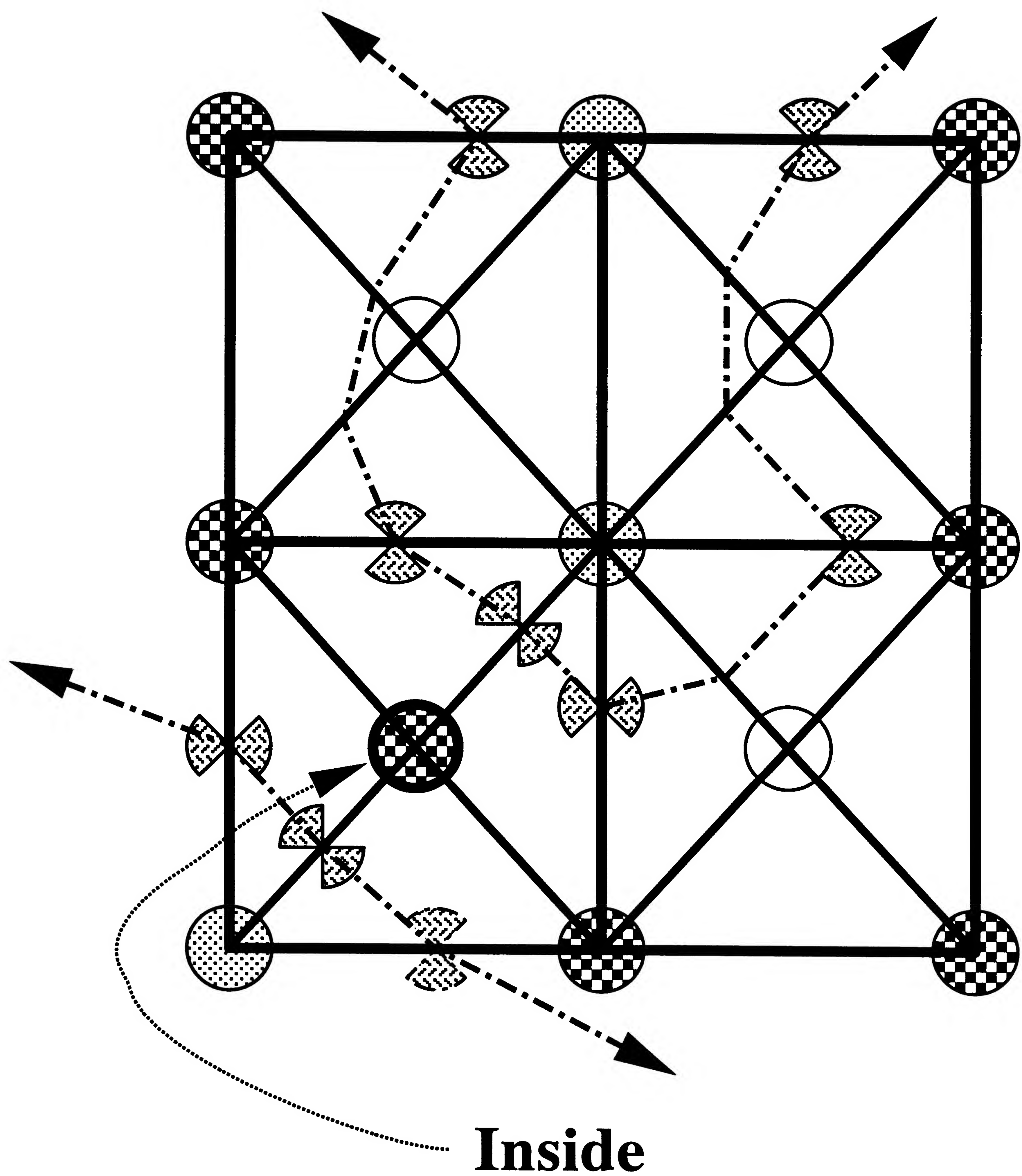


Figure 3.15: One of two possible topologies that depend on the new vertex state.



Figure 3.16: The other two possible topologies that depend on the new vertex state.

### **3.1.1.15 Other shape boxels**

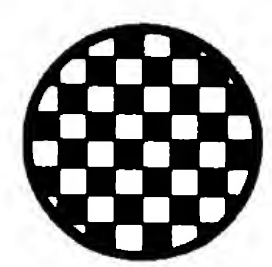
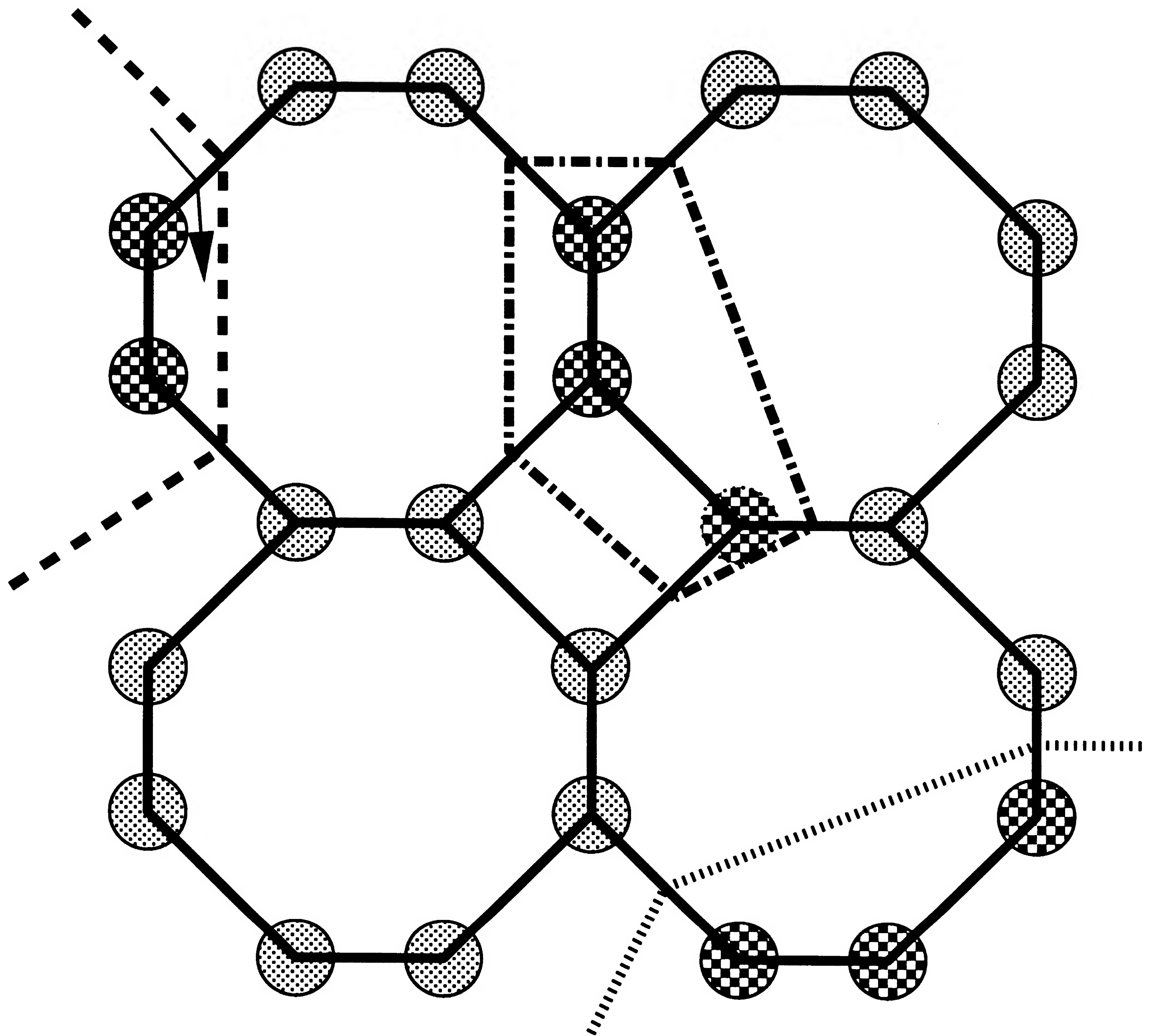
An interesting property of SpiderWeb B is that it can build contour lines in any shape voxel. Figure 3.17 shows a combined octagon and square tiling, and by following connected faces, paired hits, and knitting high, we can build contour lines through this irregular lattice. In the three dimensional case, we might have a wedge shaped fillet geometry element, joining a cuboid voxel, and then a tetrahedral element produced by an adaptive mesh generator. These rules could be used to build case tables, or the lines could be directly knit using this algorithm. A major problem with adding interpolated vertices and edges is that the computational load and memory demands become objectionable. In three dimensions, it becomes even more of a problem even on high performance computer graphics workstations.

## **3.1.2 2D SpiderWeb curve properties**

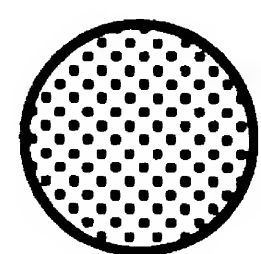
### **3.1.2.1 2D Curvature Limitation with respect to Edge Length**

The minimum curvature of a contour line is limited by the length of a boxel edges. Figure 3.18 shows that any turning radius that is less than an edge length will never appear, because it would imply two hits on an edge, which is not possible. Therefore we can estimate that the minimum resolvable radius of curvature is less than two edge lengths, by spatial analog of the Nyquist sampling theorem [112] [113] and by Shannon information theory [114]. The minimum radius of curvature in a two dimensional contour line must be greater than the boxel length in that direction. We call this radius the Nyquist length.

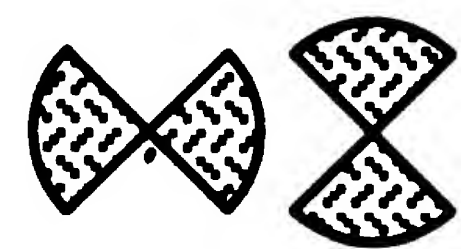




**Inside Pixel**



**Outside Pixel**



**Hit**

Figure 3.17: Other boxel shapes.

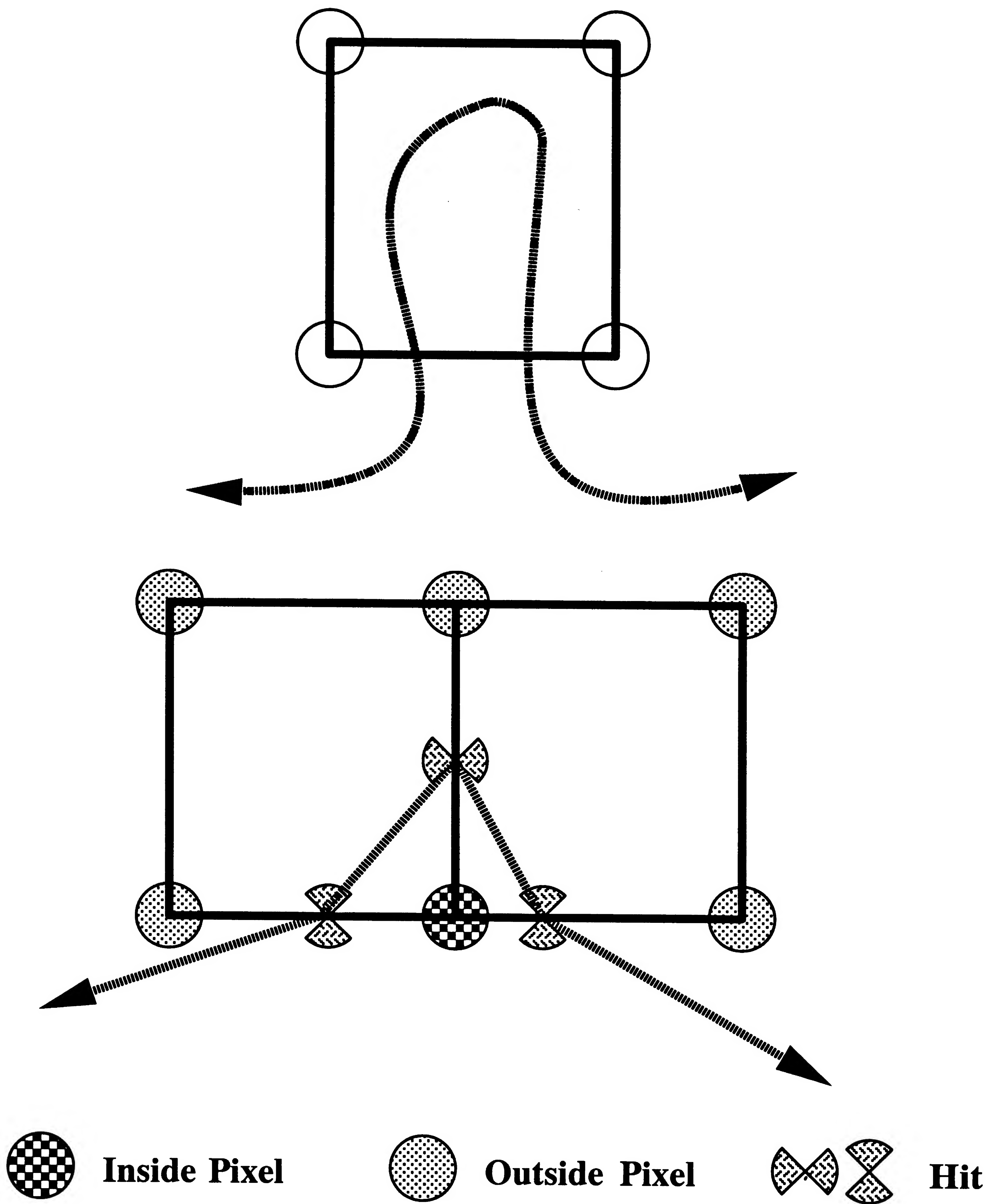


Figure 3.18: Minimum U-Turn radius as a function of edge length.

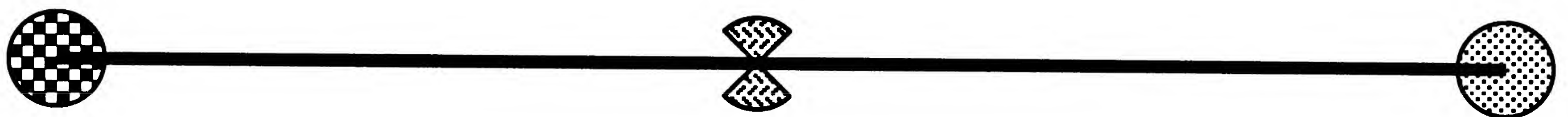
### 3.1.2.2 2D Gradient strength

The same position for a hit can be produced by different values. The **strength** of a hit is the magnitude of the gradient that produces the hit. A hit, and the consequent line or surface that is produced will have the same geometry, but the gradient that produces the geometry can be widely different. Geometry produced by a weak gradient will be more noisy, characterized by 4-hits on a face. Additionally, the degree of confidence that can be attributed to the geometry is less than the confidence that can be placed in geometry derived from a strong gradient. A strong surface is one derived from a strong gradient. The strength of the gradient is the difference between the inside vertices and the outside vertices across the surface, dotted with the surface normal. Figure 3.19 gives two example hits along an edge, one with a weak gradient,  $\delta = 1.0$ , the other with a strong gradient  $\delta = 120.0$  units. Figure 3.46 shows preliminary three dimensional surface results showing experiments to illustrate the relationship between gradient strength and bubbles. The color of the surface is determined by the difference across the inside to outside voxel vertices. The thick (green) lines are non manifold bubble triangle edges.



## Isosurface Threshold=60.0

**Weak Gradient,  $\Delta = 1.0$**



$$V_{i,j} = 60.5$$

$$X = 0.0$$

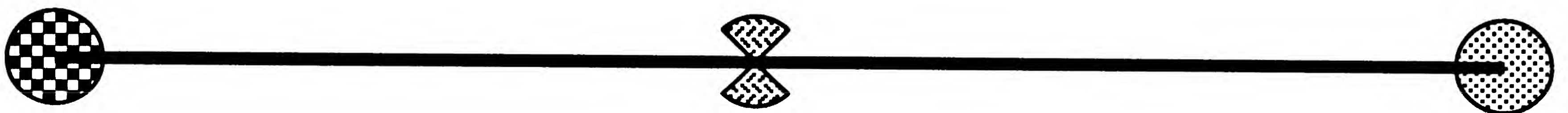
$$V_{hit} = 60.0$$

$$X = 0.5$$

$$V_{i+1,j} = 59.5$$

$$X = 1.0$$

**Strong Gradient,  $\Delta = 120.0$**



$$V_{i,j} = 120.0$$

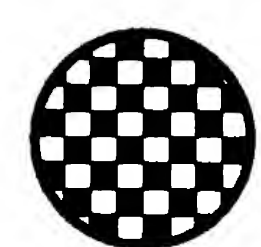
$$X = 0.0$$

$$V_{hit} = 60.0$$

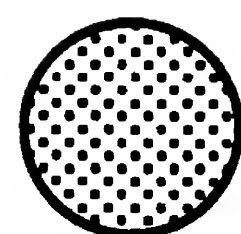
$$X = 0.5$$

$$V_{i+1,j} = 0.0$$

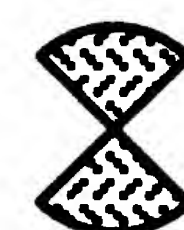
$$X = 1.0$$



**Inside Vertex**



**Outside Vertex**



**Hit**

Figure 3.19: The position of a hit is independent of the gradient strength.

### 3.1.2.3 2D Noise

Noise randomly perturbs the position of hits in a contour line plot or surface reconstruction. It roughens the surface, and when the intensity of the noise decreases the radius of curvature of the surface perturbations to the Nyquist length, 4 hits on a face occur. This causes bubbles in SpiderWeb A surfaces. Surfaces produced in weak gradients are also more likely to break up into bubbles than surfaces produced by a strong gradient. Noise that produces bubbles in SpiderWeb A is considered to reflect a fundamental undersampling of the data with respect to the Nyquist length.

### 3.1.2.4 Jordan Separation and Curve Theorem

The Jordan separation theorem [115] and the Jordan Curve Theorem [116] [117] have important implications for contour lines and isosurface sheets [69]. Figure 3.20 shows that if we consider each line segment as a sheet, only an even number of sheets can converge at a point. We show that an odd number of sheets would create a contradiction. If there were three sheets (Label A) meeting at a point, we would have a contradiction as to which side of a sheet was **In** or **Out**. Therefore, a Jordan surface can only split into an even number of component sheets. The point where they meet is labeled the articulating point, and the implication of this additional point will be made clear later in the three dimensional triangulation situation. We will show later in the discrete case, that there can only be an even number of hits on a voxel face, analogous to what we show here for the continuous situation. In the case of a contour lines, each vertex in a line can only be shared by an even number of edges. For a surface tiled with triangles, each triangle edge can be shared by an even number of triangles. An isosurface can never cross itself. It can only osculate (kiss) at a point. This point may be represented by a singular face in the discrete domain (see Section 3.2.5). This point has many properties analogous to the articulating point described in section 3.2.2.3.

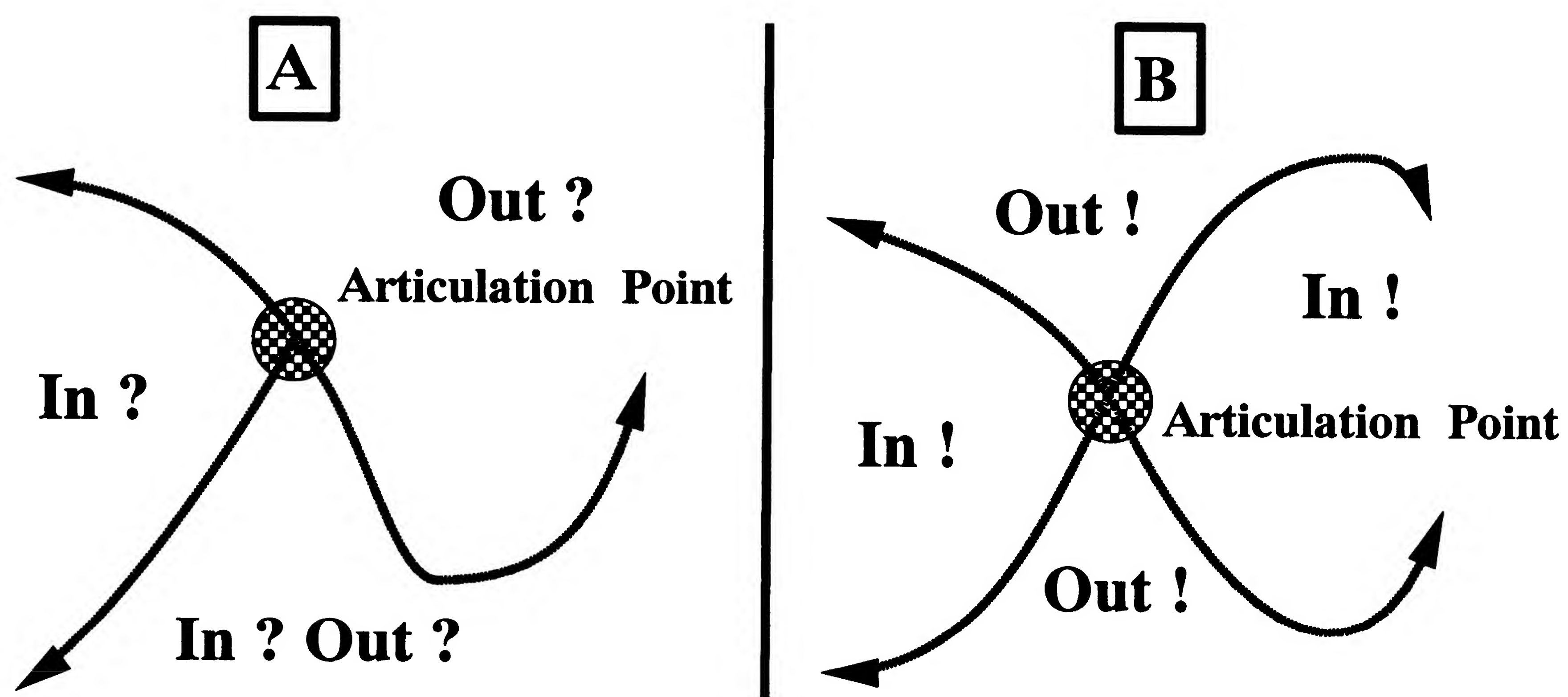


Figure 3.20: A Jordan curve separates inside from outside.



### 3.1.3 Uses of contour lines to preview 3D reconstruction

SpiderWeb isovalue contour lines are an important part of preparing a 3D SpiderWeb isosurface rendering. The contour lines in a plane represent the intersection of the SpiderWeb surface with the contoured plane. Fast planar isocontours provide an interactive view at the level of the slice of the volume geometry produced by the SpiderWeb algorithm. The user interactively mouses a threshold value slider to select a value that produces smooth, bubble free lines. The value that produces a smooth system of lines also is the value that cleanly segments the volume. However, variations in the base image level from slice to slice make it difficult to precisely predict what threshold value will cleanly segment and give “nice” surfaces for an extended range of slices. A nice surface is an continuous, coherently oriented, and unbounded surface. Figure 3.1 shows the contour lines produced by the PixelPicker window of an MRI slice thresholded at a  $\tau = 134$ . Figure 3.2 shows a magnified image of a region in Figure 3.1, showing individual boxels that frame the isocontour lines. A pattern of 4 hits on a boxel face can be seen, creating a 2 dimensional bubble characteristic of regions of weak gradient or noise. If a particular isovalue is found to produce relatively clean lines in the plane, this value is used to contour surfaces in the much slower SpiderWeb section of the program. A clean system of lines or surface is one that has few or no 4-hits on faces. Values that give clean two dimensional lines also give clean segmentation surfaces. Note the same value of disconnected lines can represent different anatomic features, such as the scalp/air interface, and the brain/cerebral spinal fluid interface. We show later that by tracking one system of surfaces, we can ignore same value, disconnected surfaces to obtain clean segmentation of same threshold objects.

### **3.1.3.1 Summary of SpiderWeb A and B for 2D**

- To summarize for SpiderWeb A, when recursing through voxel faces following hits, we consistently choose to knit all the hits in a face into possible bubbles as a representation of the algorithm's unwillingness to make a possibly wrong knitting decision. Bubbles may represent a fundamental ambiguity in the data. SpiderWeb A hedges its bets.
- To summarize for SpiderWeb B, when recursing through edges following hits, we always consistently choose to follow the boxel edge to the high, inside vertex, and continue along the face edges until we reach the next hit to complete the pair. SpiderWeb B chooses consistently to follow the path of a water stream line over the data surface.

## 3.2 Spinning Three Dimensional Surfaces from Volume Data

In this section we generalize from the construction of isovalue contour lines in a slice to spinning an isovalue surface triangle mesh in a volume. We shall describe the SpiderWeb algorithm for three dimensions which generates a packed, simply connected triangular tiled surface, or tessellation[118]. This tiled surface can accurately and precisely (to within a voxel length) best reflect the underlying continuous topology [119] [120]. SpiderWeb A does not make a decision in the case of four hits on a face, and we “hedge” our bet and make a bubble. SpiderWeb B makes an assumption to enclose the “inside” vertex. The use of an additional “articulating point” disambiguates triangle spinning. These two approaches for dealing with ambiguity are the essence of the SpiderWeb algorithm. The issue of the creation of co-planar triangles is discussed.

First, we will shall explain some some basic geometrical and topological properties we will build upon to spin triangles in three dimensions. The requirements for the data are that the underlying density function is continuous, smooth, and bounded by zero density. These constraints were introduced in Section 3.1.1.1 and are reviewed here to make this section self-contained.

Next, in order to prove that the triangles we knit in voxels never have gaps or holes, we introduce two concepts unique to 3D contours. The first is the “voxel facewise continuity rule” and the second is the “articulating point”.

The Voxel facewise continuity rule specifies that every triangle knit in a voxel face must first cross the voxel face. In this way, every triangle will have an edge on a voxel face that will abut with the triangle in the opposite face on the adjacent voxel. This idea is illustrated further in Section 3.2.2.2.

The articulating point is an additional point that is the center of mass of the connected system of hits we are building. The two remaining edges of each triangle are knit back to this point. All triangles are knit by consistently circulating triangle edges across the voxel face, then inside the voxel to the articulating point, then emerge to the voxel edge hit. A more detailed explanation is given in Section 3.2.2.3.



### 3.2.1 Three Dimensional Surfaces in Volume Data

We shall show in this section that the development of the properties of a contour line from boxels is directly applicable to the generation of triangle mesh surfaces from voxels. The only new element to make the three dimensional analysis possible from the two dimensional analysis is the addition of an additional disambiguation point, or points, to the connected system of hits on a voxel face. Each connected system of faces and edges is joined not only by the arguments given above, but in the three dimensional case, by an additional point that makes the analysis of each voxel face independent of each other. Thus, our analysis of voxel derived surfaces follows directly the analysis of boxels. As we shall see, triangles are knit from on the faces of the voxels and the articulating point. SpiderWeb contour lines are also used to preview three dimensional SpiderWeb geometry as it intersects the slice plane. In some ways, the SpiderWeb surface resembles embedded triply-periodic minimal surfaces produced by soap films on cubical lattices, as analyzed by Schoen [121].

In order to build a conceptual framework for triangle mesh generation, we first look at the properties of the underlying function. We make only three mild assumptions about our volume data. We make one convention for orienting inside/outside. This section recites the properties of two dimensional functions specified in Section 3.1.1.1 to make this section self-contained.

- **Smoothness.** The density function must be smooth. This is discussed with reference to the two dimensional SpiderWeb algorithm in Section 3.1.1.1.
- **Continuity.** The density function must be continuous, *i.e.*, there are no singularities in the volume. We always assume that the samples are of a continuous function. This is discussed with reference to the two dimensional SpiderWeb algorithm in Section 3.1.1.1.
- **Boundedness.** The volume must be bounded by a zero density. In any direction, when approaching infinity, the density value is zero. We can not have an isosurface that is a hyperbolic sheet, extending out to infinity. All of our surfaces must enclose a finite volume and have a finite surface area. Again this is discussed with reference to the two dimensional SpiderWeb algorithm in section 3.1.1.1.

Our orientation convention is that the “Outward” or “Outside” direction is toward the lower density value. Outside is always at infinity, and is at a lower, or zero density with respect to Inside.

From the previous specification of our smooth, continuous, and bounded volume density function, we can see that there are no possible breaks or holes in the surface, no matter how closely you examine the surface. There can not be any loose edges of our isosurface, unconnected to another bit of surface.

### 3.2.2 Three Dimensional Voxel Definitions

We assume that our volume density data is sampled in a three dimensional lattice or array, indexed by  $i, j, k$ , bounded by  $0..I, 0..J, 0..K$ . Each **slice** consists of the subset of the array  $0..I, 0..J$  at a particular  $k$ . Each element of the slice array is a **pixel** at index  $i, j$ . Each pixel is a sample value at an integral coordinate value. A **voxel** is constructed from 8 adjacent pixels, spanning  $i..i+1, j..j+1, k..k+1$  for any particular voxel indexed  $i, j, k$ . Thus a **voxel** refers to the region of space bounded by pixels at the 8 **vertices** of the cuboidal voxel. Each voxel consists of 8 voxel vertices, 12 **voxel edges**, and 6 **voxel faces**. Each cube edge is a voxel edge. Each lattice point is a voxel vertex. Each voxel edge consists of the exterior path between two exterior voxel vertices. To insure the boundedness of our data, each data array dimension is prepended and appended with zero values at all 6 surfaces of the data box.

For any arbitrary threshold value, each of the 8 vertices can have two states. The state of a vertex is **Inside** or **Outside**. **Inside** is defined by our convention as having a higher density than the threshold. **Outside** is defined as having a lower density pixel value than the threshold. At an infinite distance, we assume a zero density, and an **Outside** orientation. **Inside** is less than or equal to some threshold value; **Outside** is strictly greater than the threshold value. This effectively bounds the data with a zero value border.

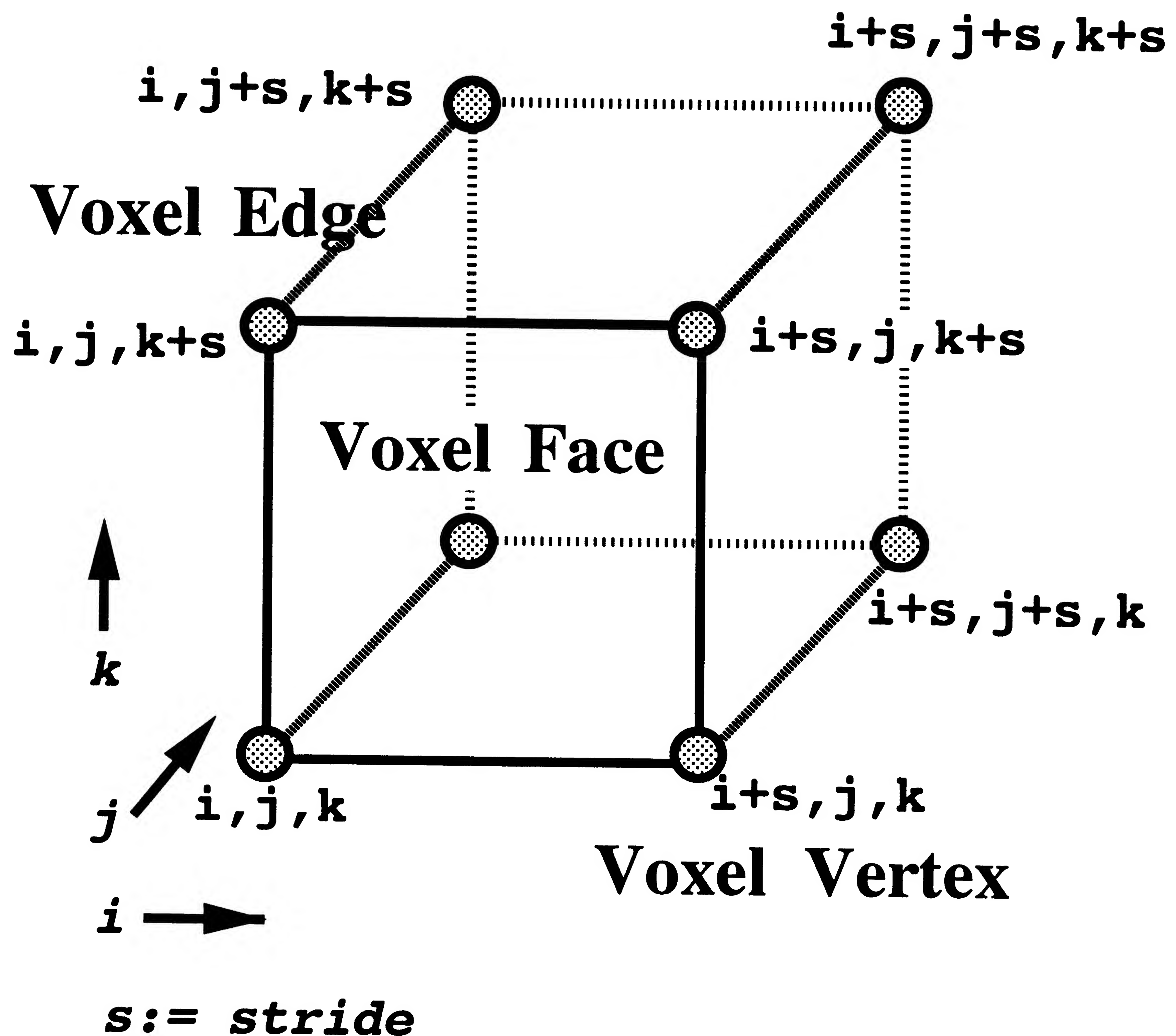


Figure 3.21: A voxel consists of a cuboidal arrangement of data lattice edges. A **voxel vertex** is an element in the data lattice. A **voxel edge** is the exterior path between to coplanar voxel vertices. A **voxel face** is a planar arrangement of exterior voxel edges. The **stride** is the step through the data lattice to construct voxels. Frequently, we do not want all the data in the data lattice. For a stride of 1, we would sample every other data element in the data lattice.



Voxels are packed together in the data lattice. Every element in the data lattice is used in multiple ways. A data element is a voxel vertex and a voxel face vertex. The edge between a two data lattice points are used as voxel edges, voxel face edges. If there is a hit along that edge, it is also used as a triangle vertex. Each voxel has an adjacency relationship with the surrounding voxels. Each voxel shares various constituent parts with adjacent voxels. The basic definition of an adjacent voxel is a voxel that shares a voxel vertex, a voxel edge, or a voxel face. A voxel that shares a vertex with a neighboring voxel is said to be vertex adjacent voxel. A voxel that shares an edge with another voxel is an edge adjacent voxel. A voxel that shares a face with another voxel is a face adjacent voxel. Figure 3.22 illustrates three pairs of voxels. Each pair is adjacent by their shared vertex, edge, or face. There are 8 possible vertex adjacent voxels. There are a total of 12 possible edge adjacent voxels. There are a total of 6 possible face adjacent voxels. Voxels can be considered adjacent by various combinations of vertex, edge, and face adjacency. The union of vertex, edge, and face adjacency is known as  $(12 + 6 + 8)$  26-adjacent. Face and edge adjacency is 18-adjacent. Udupa [106] and Kaufman [122] use the faces of adjacent voxels to directly tile surfaces. They cite that different types of these (18-, 26-) connected surfaces use different voxel adjacencies.

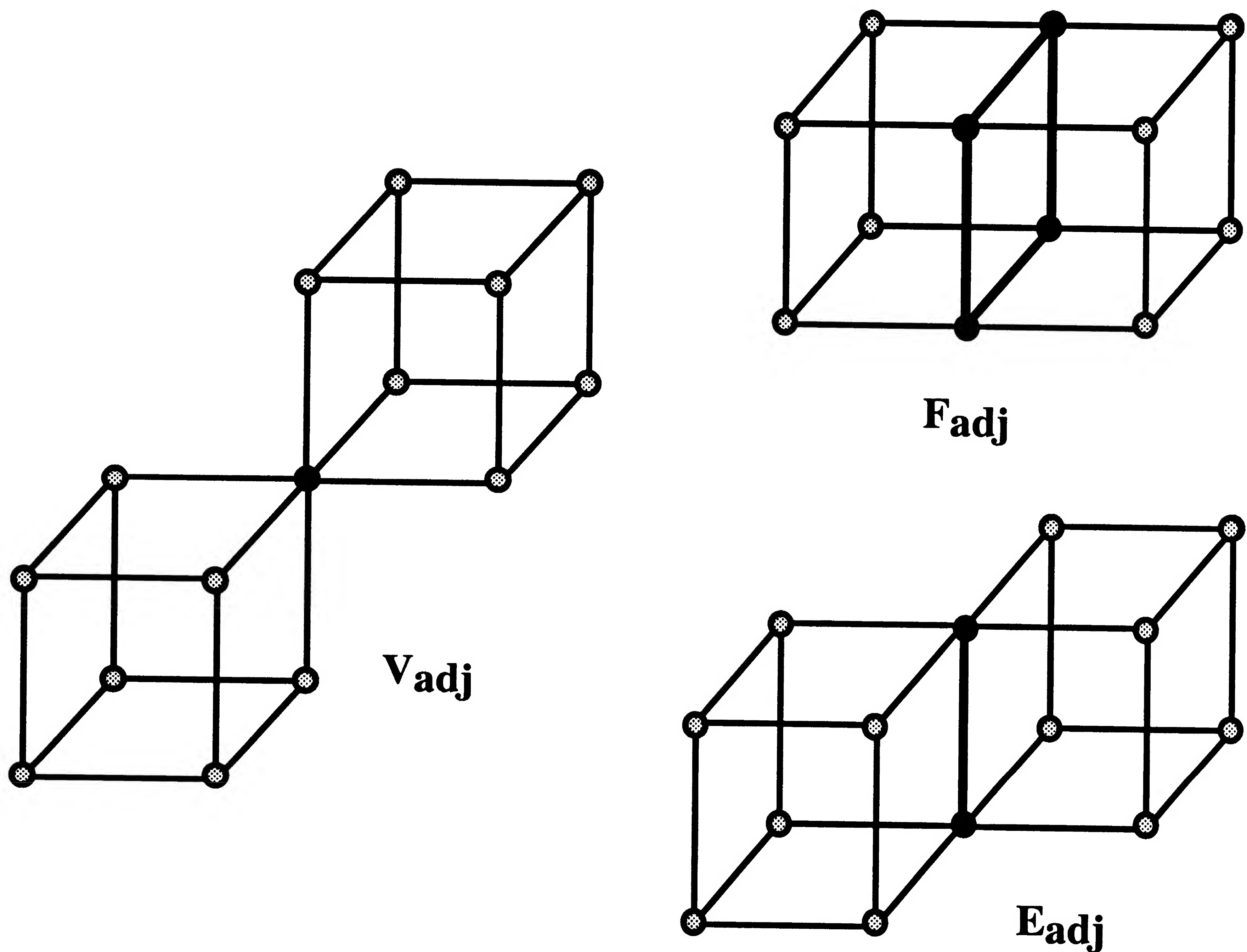


Figure 3.22: **V<sub>adj</sub>** are vertex adjacent voxels that share a common vertex. **E<sub>adj</sub>** are edge adjacent voxels that share a common edge. **F<sub>adj</sub>** are face adjacent voxels that share a common face. There can be only 1 face adjacent voxel, 3 edge adjacent voxels, and 7 vertex adjacent voxels. A face adjacent voxel must share 4 vertices. A face adjacent voxel must share 4 edges. An edge adjacent voxel must share two vertices.

The definition of a hit for voxels is strictly analogous to that of boxels. An isovalue threshold **hit** on a voxel edge in three dimensions is given by linear interpolation along the voxel edge. For a voxel edge  $Ve$ , consisting of voxel vertices  $V_0$  and  $V_1$ , with positions  $V_{x0}, V_{y0}, V_{z0}$  and value  $V_{v0}$ , and  $V_{x1}, V_{y1}, V_{z1}$  and value  $V_{v1}$ , there exist 4 states with respect to a threshold value  $\tau$ :

$$(3.7) \quad (V_0 > \tau) \ \&\& \ (V_1 > \tau)$$

There is no hit on  $E$ .

$$(3.8) \quad (V_0 < \tau) \ \&\& \ (V_1 < \tau)$$

There is no hit on  $E$ .

$$(3.9) \quad (V_0 < \tau) \ \&\& \ (V_1 \geq \tau)$$

$$(3.10) \quad (V_0 \geq \tau) \ \&\& \ (V_1 < \tau)$$

There is a hit on  $E$ . The position of hit  $E_{h\{x,y,z\}}$  is given by:

$$(3.11) \quad E_{hx} = V_{x0} + (V_{x1} - V_{x0}) \frac{\tau - V_0}{V_1 - V_0}$$

$$(3.12) \quad E_{hy} = V_{y0} + (V_{y1} - V_{y0}) \frac{\tau - V_0}{V_1 - V_0}$$

$$(3.13) \quad E_{hz} = V_{z0} + (V_{z1} - V_{z0}) \frac{\tau - V_0}{V_1 - V_0}$$

We observe again that there can be only one hit along any voxel edge.

### 3.2.2.1 Properties of Hits

Before we define the knitting rules for hits, we will define some properties of hits that are independent of how we connect (knit) them into a sheet of triangular tiles.

Any voxel edge that has a hit connects the two voxel faces that share that hit. These faces are **connected faces**. This is because hits must always come in pairs, and if a face has a hit, the adjacent face that shares that edge will also have a hit. The faces are connected by the edge hit pair requirement.

The importance of the concept of connected faces is made clear in recursive scanning of the connected components of a sheet. Consider starting a depth-first search at some seed voxel face. Figure 3.23 illustrates a voxel that we enter into via



Face 1. Let us walk through this voxel and examine each step. We start entering the voxel on edge A on face 1.

We know that there must be another hit on Face 1 (because hits must occur in pairs). Our entry edge and the hit on that edge A is illustrated as hit 1a. However, Edge A is shared by Face 2, so we know that we must scan Face 2 for an additional hit. If an edge in our scan path does not have a hit, then there is no need to scan the voxel face that shares that edge.

We can consider Face 1 and Face 2 connected by shared Edge A hit 0, as shown in Figure 3.23. In a cuboid voxel, a voxel can be visited at most three times if a system of disconnected faces (and edges) containing hits is in that voxel. Figure 3.24 illustrates two disconnected systems of hits on a voxel.

Consider a system of hits tracked on the first entry of the SpiderWeb into this voxel. If the algorithm entered the voxel on the left face, it would find two hits,  $H_0$  and  $H_1$ . It would then recurse into the faces that share these hits. If we recurse along the  $H_0$  edge into the front face, we would see hit  $H_2$ . That would take us into the bottom face, and back to  $H_1$ .

Since each of the faces are now visited, the algorithm exits the voxel after spinning the triangles associated with these hits and this articulating point. This articulating point consists of the mean position of  $H_0, H_1, H_2$ , as is explained in Section 3.2.2.3. It is not shown in this diagram to keep the figure uncluttered.

Now, should another surface, or the same surface enter this voxel, say from the top face, it would find  $H_3$  and  $H_4$ . Recursing into the back face it would find  $H_5$ , and then enter the right face. This, separate surface would be complete, and the algorithm would exit after computing the articulating point of  $H_3, H_4, H_5$ , and spinning triangles. As long as there are no faces containing hits that have hit edges in common between each face, the surface tracking SpiderWeb algorithm will not touch that surface again. This is true even if the two surfaces pass each other within the body of a voxel. In this way the voxels are re-entrant.

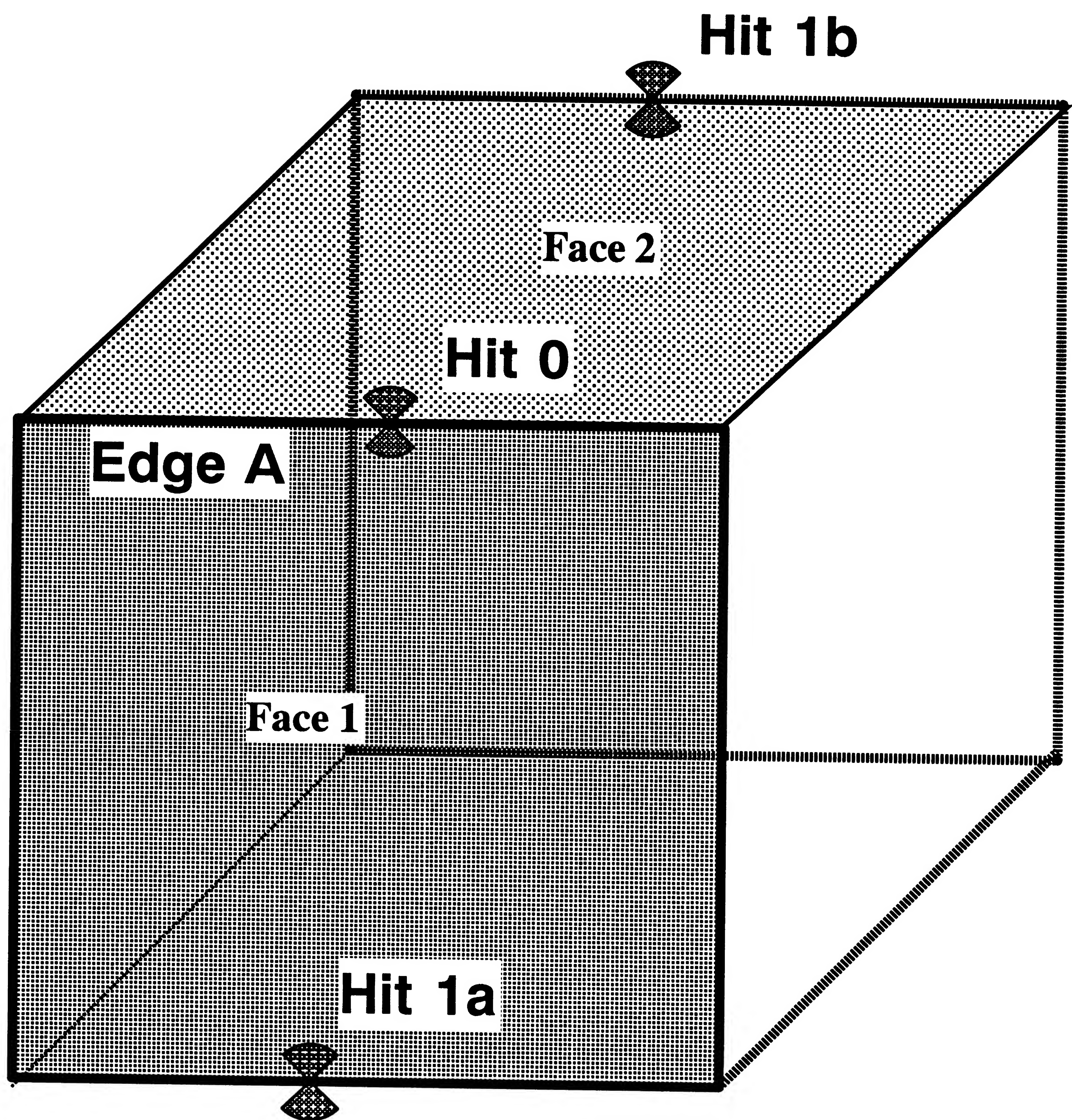


Figure 3.23: Voxel faces are connected by a shared edge with a hit.



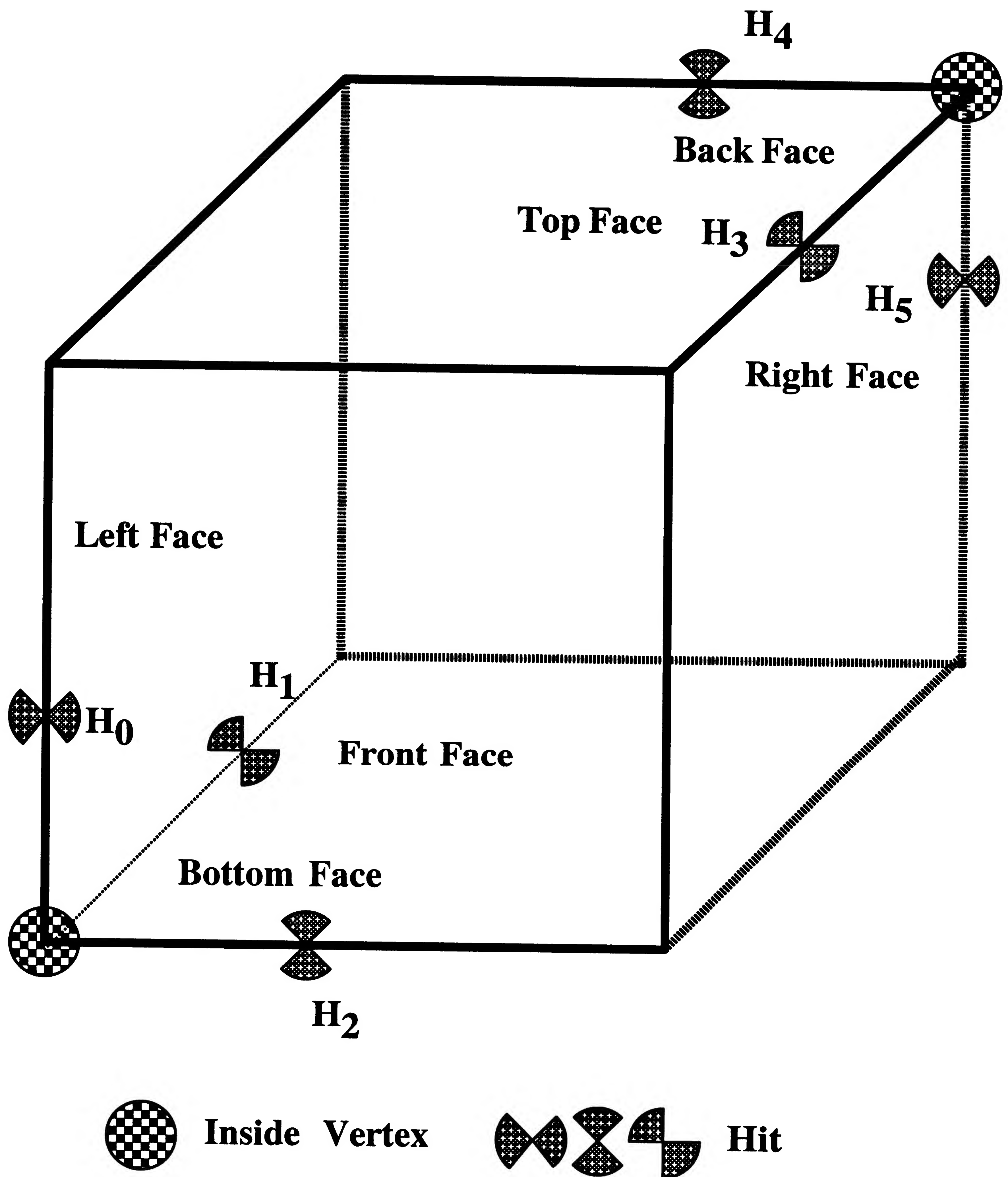


Figure 3.24: Two disconnected connected systems of hits passing through a voxel.



Within each voxel face that contains at least one pair of hits, those edges containing, and between, a pair of hits are said to be connected. The importance of the concept of **connected edges** can be seen when depth-first recursing along a surface. When tracking a surface, we constantly must knit on new triangles from hits. As we find the first hit of a pair, we must scan along voxel edges for the hit that will close the pair. As we accumulate hit pairs, we can calculate the articulating point and spin triangles. Each edge that is traversed in this way is said to be connected. Also, each edges so traversed can be considered to be owned by a particular connected system of hits. Edges are connected by pairs of hits. All edges between and possessing a pair of hits is said to be connected.

Any system of hits that is in a connected face and in a connected edge is considered to be connected hits. Thus, we can group hits as belonging to a surface without explicitly tileing that surface in any way.

### 3.2.2.2 The Voxel Face-Wise Continuity Rule

We require that any surface tileing we construct must not have any holes in it. In tileing nomenclature is must be packed[118]. There must not be any holes or gaps in the triangle tiles between voxels, and within voxels. In this section, we will look at defining a rule that will guarantee no possibility of holes between voxels. For any triangles we knit inside a voxel, we specify that at each voxel face that contains triangles, the same triangle's edge is shared by two triangles on the inter-face between voxels. Equivalently, we can say that any pair of triangle edges that are in congruent voxel faces must have congruent triangle edges. The difference is the nomenclature system we use to identify lattice points, voxel vertices, voxel face vertices, and finally, triangle vertices and triangle edges. We are referring to the same locations in with different aliases. More simply put, what goes in at one voxel face must come out on the adjacent voxel, opposite face (the voxel face that is congruent, or touching). At this point in our development of the three dimensional triangle spinning rule, we have not specified any rule or method for spinning triangles. Our goal is to develop a condition for guaranteeing surface tile continuity between voxels. In the next Section, 3.2.2.3, we will show how to guarantee triangle surface tileing continuity within a voxel with the Articulating point.

Figure 3.25 shows two voxels, labeled A and B. They share a common voxel face formed by the voxel vertices  $A_0, A_1, A_2, A_3$  from voxel A, and the voxel face formed by the adjacent voxel B. The B voxel face is denoted by the cycle through

voxel vertices  $B_0, B_1, B_2, B_3$ . The voxel vertices  $A_0$  and  $B_0$  occupy the same location in space, and indeed, are the same “thing”. The voxel edge  $A_0$  to  $A_1$  occupies the same space as voxel edge  $B_0$  to  $B_3$ . Both of these objects occupy the same space, but have different ownership by voxel A or B. They are said to be congruent. The voxel faces shown in Figure 3.25 are congruent. For the purposes of illustration, I have teased the voxel faces and edges apart, and I have separated the triangle edges that lie in the voxel faces.

The triangle in voxel A has an edge that lies in voxel face  $A_{0,1,2,3}$ . The triangle in voxel B has an edge that lies in voxel face  $B_{0,1,2,3}$ . Voxel A and B are adjacent, which means they share a common face. Equivalently, they each have a congruent face, owned separately. The relationship between adjacent voxel shared faces is called complementary. This is because the direction of circulation (order of traversal) is opposite from one congruent face with respect to the other. Clockwise from the point of view inside voxel A is opposite relative from the indexing scheme of voxel B. This can be seen in the indexing of voxel vertices from voxel A and voxel B. Vertex  $A_0$  is congruent to vertex  $B_0$ , and vertex  $A_2$  is congruent to vertex  $B_2$ . However, vertex  $A_1$  is congruent to vertex  $B_3$ , and vertex  $A_3$  is congruent to vertex  $B_1$ . This shows the opposite ordering of mirrored, and congruent voxel faces.

Consider the 6 voxel faces. A cyclic number ordering of the voxel faces, 0, 1, 2, 3, 4, 5 would be the same in each voxel. When we consider the voxels assembled into a lattice, we can see that adjacent faces can not have the same index. If we use symbols for voxel faces, we can describe a voxel as having a top, bottom, near, far, left, and right voxel face. If we stack up some voxels like building blocks, it is clear that we can never have two voxel north faces adjacent to each other.

Within each voxel face, we establish an ordering of the voxel vertices. For each voxel face, from a point of view inside the voxel, looking out from that face, we can index voxel face vertices in a clockwise (or counterclock wise) order. The sense of the ordering is not important as long as we consistently apply our ordering convention. We can consider each voxel face to have a top, bottom, left, and right vertex. From the voxel perspective, we will be indexing the same voxel vertices from different faces with different indices, or names. Our overall plan is to establish a voxel face relative ordering so that the rules we have developed from two dimensional boxel contour lines can be directly applied in the three dimensional voxel setting. When we index voxel face vertices in a conventional cyclic order, we realize that the adjacent voxel face vertices are locally referenced in a complementary, or mirrored order relative to our local voxel face vertex ordering.



In order for there to be no holes in our tiled surface sheet, wherever the mesh of triangles passes into a voxel, it must exactly match the edges of the triangle tiles passing into an adjacent voxel. More specifically, the triangle mesh passes continuously into the adjacent voxel complementary face. This continuity is insured by the congruence of the shared hits on the voxel face edges, and the use of a consistent triangle spinning rule in each voxel face. At the voxel face we spin or knit triangles, either by a case table lookup or a SpiderWeb rule based algorithm. If every triangle that has an edge that lies in a voxel face has a congruent triangle edge in the adjacent voxel complementary face, this surface can not have artifactual holes or tears at the voxel interfaces. This condition is called the voxel face-wise continuity rule. If we use the same rule to spin triangles in each voxel face, then the triangles knit from shared voxel faces and shared voxel edge hits will always match up at the voxel faces. Triangles knit from these shared hits with the same rule will have congruent voxel face triangle edges. From this, we can easily see that holes are impossible if this condition is obeyed.

When building a tileing case table we must be careful to observe the voxel face-wise continuity condition. A cuboid voxel with 8 vertices with 2 possible values (inside or outside) can have 256 possible cases. For the  $2^8 = 256$  possible voxel hit combination cases, the possible pairs of voxels is  $(2^8)^2 = 65,536$ . This rule applies for each possible pair of voxel cases  $((2^8)^2 = 2^{8*2})$  and for each possible pair  $(6^2)$  of adjacent faces. When we take into account each of the 6 voxel faces for each voxel case pair combination, we have  $2^8 \times 6 \times 2^8 \times 6 = 2,359,296$ . There are 2,359,296 combinations of 256 possible voxel cases taken for each of 6 faces, paired by case and face. These additional cases of voxel face pairs are called extended voxel cases. A large number of these extended face pair cases are for voxels or voxel faces with no triangles. Significant simplifications can be made, but this explosive growth of adjacent cases explains why implementations of “marching cubes” type algorithm lookup tables can have maddening problems with tileing errors and holes. An example [123] of tileing errors can be seen in Figure 1.2.



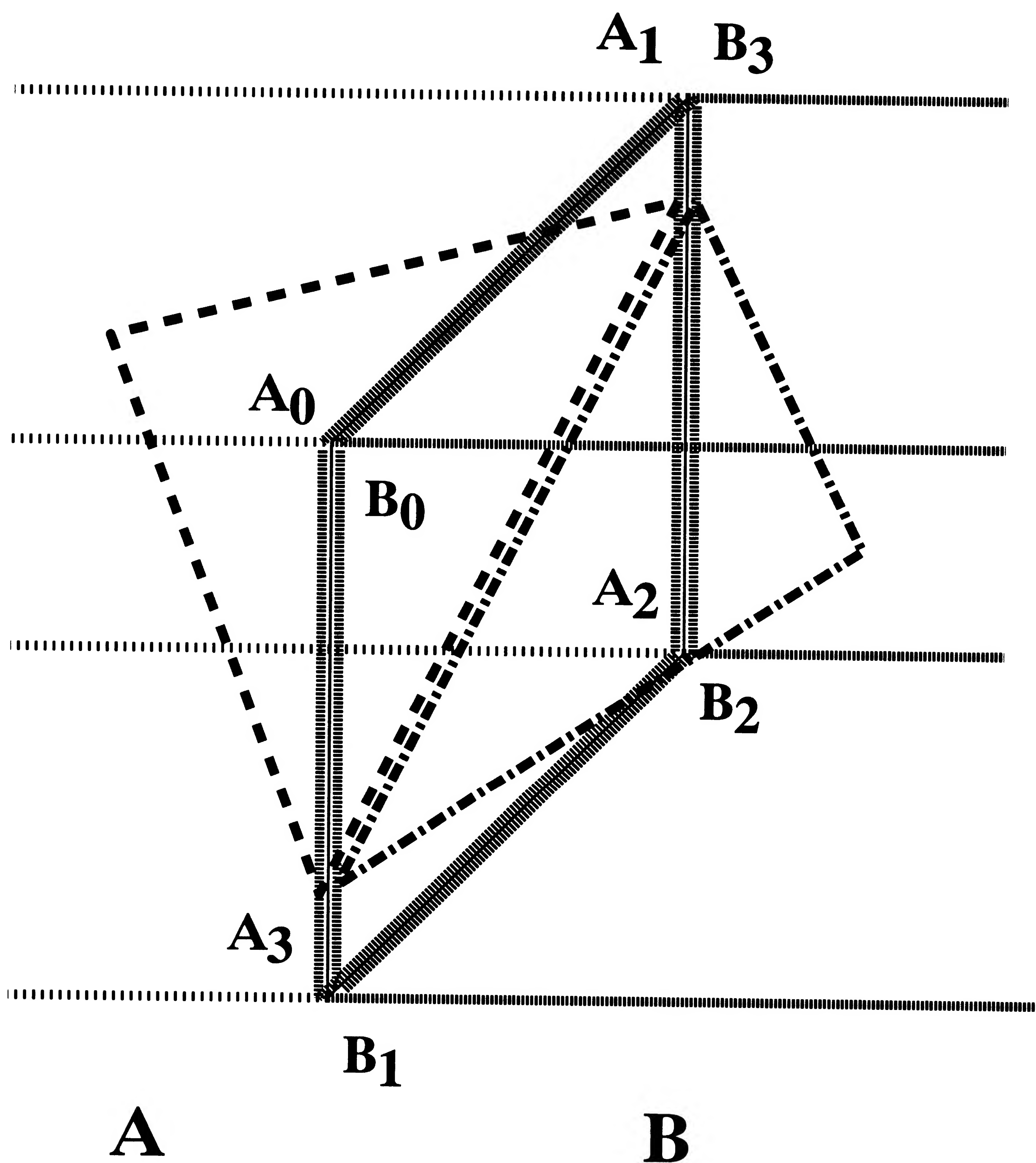


Figure 3.25: Triangle edges on voxel faces must be continuous with triangle edges in the adjacent voxel opposite face (“voxel face-wise continuity rule”).

### 3.2.2.3 The Articulating Point

As we have seen for the continuous surface in Figure 3.20, an isosurface can meet at a point only if there are 2 or 4 converging component parts of the sheet. The sheet can only “evenfurcate”. The sheet can not trifurcate. In tiling this sheet with triangles, we will introduce an additional point interior to the voxel. This **articulating point** simplifies the algorithmic creation (spinning) of a triangular tiling for all possible voxel hit cases. Wyvill [59] also used the voxel face hit average value with bilinear interpolation to disambiguate “hard” cases. The articulating point is the mean (centroid) of the system of connected hits in a voxel. It is possible to have multiple articulating points when a separate sheet re-enters a voxel. The articulating point is used as a vertex for subsequently spun triangles inside the voxel. For a collection of voxel edge hits  $h_n$ ,  $n = 0, 1..m$ , the position of the articulating point,  $AP_{x,y,z}$  is

$$(3.14) \quad AP_x = \sum_{n=0}^m h_x n / m$$

$$(3.15) \quad AP_y = \sum_{n=0}^m h_y n / m$$

$$(3.16) \quad AP_z = \sum_{n=0}^m h_z n / m$$

The webbing of the triangle edges spun around the articulating point gives the interior of the voxel the appearance of a small spider web, or cobweb inside the body of the voxels (see Figure 3.32). This suggests the name for the algorithm. The articulating point is also the point about which all of the bending (articulation) of a the triangle mesh sheet occurs within the voxel.

Two hits on a face, with each hit shared twice results in three connected voxel faces. Because a connected system of hits must extend to at least three connected faces that are not coplanar, the articulating point is always inside the body of the voxel, and never flush with a voxel face. For exactly three hits on a voxel, the articulating point will always be co-planar with the hits. This is due to the fact that the center of mass of a triangle is at the centroid of the triangle face. This is not true for four or more connected hits. Triangles spun with the articulating point will have continuous and connected crease lines between triangles.

#### 3.2.2.4 Some properties of a surface tiling

In the previous sections have analyzed some properties of hits without specifying how we are going to knit these hits into triangles. We can now observe some limitations of any triangularization we wish to make of the hits. We see that the Jordan surface theorem applies, the Möbius condition follows, The Euler–Poncaé relation also follows if we construct triangles properly. This is essentially the argument made for resolvable contour line radius of curvature made previously. It is not possible to resolve a surface with radius of curvature smaller than 2 voxel edge lengths. The minimum observable u-turn radius for any surface must be at least 2 voxel lengths in that direction. We can see that a voxel surface can not reenter a voxel in the same edge it exited from, because we can not have two hits on a voxel edge. It is possible to make an ellipsoid of minimum resolvable curvature for any sampling lattice. Anisotropic sampling grids frequently arise in MRI and CT, where a slice can have  $1\text{ mm} \times 1\text{ mm}$  resolution in the plane, and 3 mm resolution between planes. Before we begin to specify how we wish to knit hits, we have some lower bounds on how acutely our surface can fold at the triangle edges.

#### 3.2.2.5 The Jordan Surface Theorem

A property of isovalue surfaces given by the Jordan surface theorem [115] [69] is that a Jordan surface segments the volume into an **inside** and an **outside** region. Figure 3.20 illustrates this for a curve. The Jordan curve theorem in three dimensions is the Jordan surface theorem.

#### 3.2.2.6 The Möbius Condition

We have seen that a Jordan surface separates inside from outside. The triangular tiles that compose such a surface are planar tiles that have a normal direction associated with them. Each tile normal is, by a convention adopted in the SpiderWeb algorithm, pointing outward. The normal direction is defined by the cross product of each pair of triangle legs. There are three possible pairs of triangle legs. Because the triangle vertices define a plane, and that plane has a normal vector associated with it, the normal vector defined by each pair of triangle legs must correspond to the normal vector of the plane. The plane of each triangle has inside voxel vertices on one side, and outside triangle vertices on the other side. This is what the Jordan surface



theorem requires. The order of traversal of triangle legs to sequentially compute the equivalent cross products must be such as that each cross product point in the same direction. This is only possible if we traverse the triangle legs in a consistent order. Triangle normals are consistent if they never point to opposite sides of a triangle mesh at any adjacent triangle edges. Triangle normals are consistent if they never point in opposite directions at shared vertices in a simplicial complex triangle mesh. Each triangle (of a simplicial complex) can be considered to have a circulation direction. A coherent orientation means that shared triangle edges are never traversed (circulated) in the same directions [124]. A directed edge is a directed graph or digraph [125]. Intuitively, each edge must be a “two way street” (digraph). The circulation orientation of each triangle must cycle consistently with each adjacent triangle. The triangle normal is oriented from the circulation orientation. The normal is correct if it points outward, by convention, on the side of the face of the triangle that points to lower density value. The Möbius condition does not permit Möbius strips. The **Möbius rule** follows from consistent circulation of triangle legs and produces and correct triangle normals.

Figure 3.26 illustrates some issues in triangle circulation and normal direction. Triangle mesh A is coherently oriented, and each triangle edge has an opposite arrow from the adjacent triangle. Triangle mesh B is incoherently oriented. It would produce incorrect normals by crossing pairs of triangle legs in the order specified by the arrows. In triangle mesh C as each triangle is not traversed in a closed path.

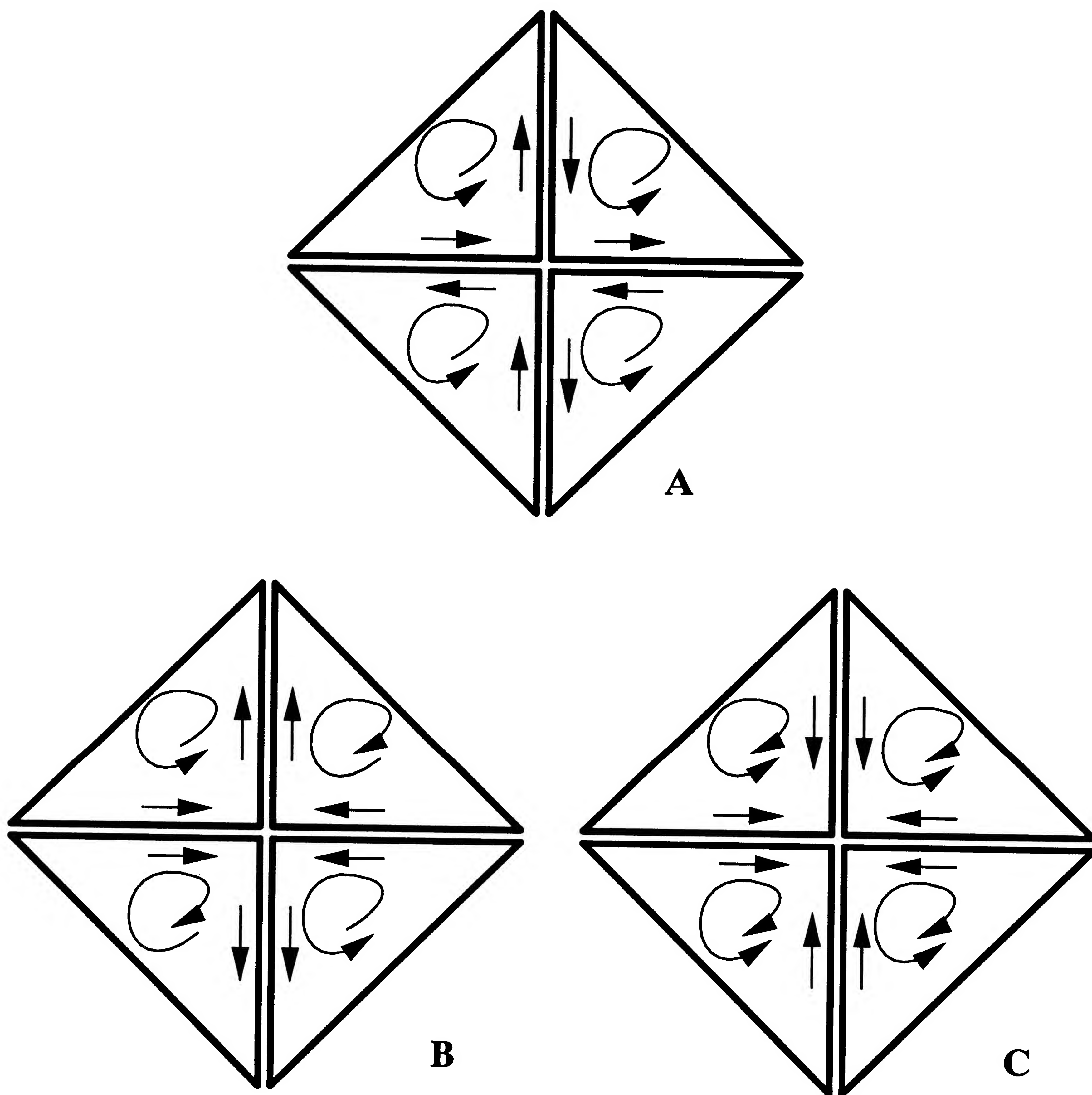


Figure 3.26: Each triangle edge must be a “two way street” (digraph) to be coherently oriented.

### 3.2.3 Triangle Definitions

A **triangle** consists of three **triangle vertices** (distinct from a voxel vertex) connected by three **triangle edges** (also distinct from a voxel edge). A **triangle face** is the plane formed by the two **legs** of the triangle. This plane is oriented, and has a normal vector that is given by the cross product with a cyclic ordered pair of triangle legs. This normal derived from the triangle defined plane is the **triangle face normal**.

The goal of the SpiderWeb is to generate a topological polyhedral representation of an object. In this section we will define some basic topological terms in the narrow setting of the two and three dimensional SpiderWeb algorithm. A **manifold surface** has a one-to-one correspondence with a flat disk. A manifold surface has no “pleats”, or locations where it splits into multiple parts. We consider a solid as a three-manifold with a boundary. We then impose a triangularization of the object for analysis and visualization. A **simplex** is a linearly independent pair of convex edges that form a triangle, for two and three dimensions. A **complex** is an assemblage of simplices, or a **simplicial complex**. A triangle mesh is also be considered as a simplicial complex. A subset of points is **triangulable** if it is homeomorphic to a simplicial complex. A triangulable set is also called a **topological polyhedron**. The **cycle** of vertices to compute the triangle legs, their vector, and the normal is called the **circulation** of the simplicial mesh. A **coherently orientable mesh** has each triangle cycle ordered in such a way as to correctly and coherently oriented triangle face normals. This means that all of the triangle face normals point consistently **Out**, according to the SpiderWeb normal convention. Contradictions in the face normal of a triangle can occur if the cross product ( $\times$ ) is not consistently calculated because the cross product is commutative.

The SpiderWeb convention for triangle normals is that the triangle face normal points outside, to the side of the face with lower density.

A triangle can have two type of normals associated with it. The **face normal** is the normal vector associated with the flat face of the triangle. When rendered, it gives a flat, tiled appearance to the triangle. Each vertex of the triangle can have a normal associated with it. This **vertex normal** is computed one of two ways. The first algorithm calculates the interpolated volume gradient at the triangle vertex. The second method determines the vertex normal as the average of all the adjacent triangle face normals.



The goal of vertex normals is to have smooth looking triangles when drawn in a modern computer graphics workstation. The normal value is linearly interpolated during the rasterization of the triangles at each pixel across the scan line. A smooth, seamless surface appearance results when the normals at each common, or shared triangle vertex are precisely the same. An example smooth, seamless surface is in Figures 6.3 and 6.4.

The face normal of the triangle is normally associated with the centroid (center of mass) of the triangle. There is no position typically associated with the triangle face normal. The SpiderWeb software can draw the triangle face normal as a short line segment  $1/3$  of the triangle perimeter from the triangle centroid. The purpose of this is to debug triangle circulation algorithms. This “hedgehog triangle normal mode” is shown in Figures 3.33 and 3.36, where the surface normals stick out of the surface like short hairy bristles. Figure 3.27 shows the triangle face and vertex normals of a mesh triangle.

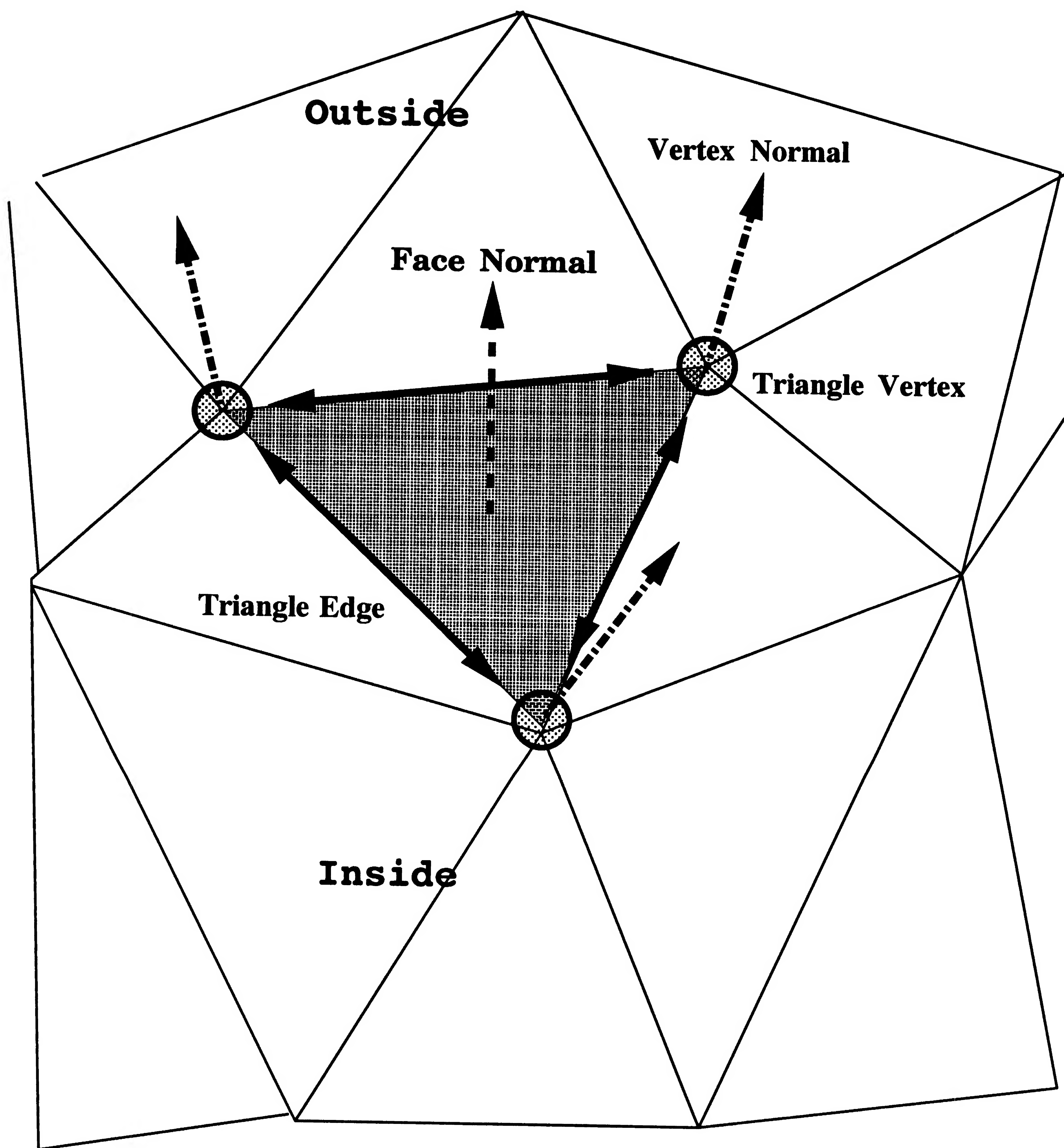


Figure 3.27: Triangle face and vertex normal vectors.

The simplest type of normal direction to assign to a particular triangle is the face normal. This gives a triangle a flat, tiled appearance. The normal value is used to calculate lighting reflectance properties, such as diffusive reflectance, specular reflectance, emissivity, and ambient light reflectance. A modern computer graphics computer calculates the color of a pixel in a triangle after the triangle has been scan-converted into a row of screen pixels[126]. The normal value associated with each pixel associated with the triangle is the same for each scan converted pixel belonging to the triangle. This gives these triangles a tiled, faceted appearance. The advantage of this type of computer graphic rendering is that one normal is specified for each of the three triangle vertices. This speeds up the rendering process. Each triangle has a circulation orientation associated with it. This gives each triangle edge or leg a direction associated with it. The cross product at each triangle vertex is taken by the ordered set of triangle legs at each triangle vertex. For a particular vertex, an triangle edge can be directed toward or away the vertex. Effectively, the vector value of an edge is signed. We cross the two legs in the circulation order. If the ordering of triangle legs is 1, 2, 3, then each cross product is generated from the cyclic ordering to yield the sets of ordered pairs 1, 2, 2, 3, 3, 1. The edges are signed in the sense that they can be considered directed edges (digraphs). A coherent orientation of triangle edges gives an orientable surface. Each of the three possible cross products will agree in sign and direction. Therefore, it is only necessary to calculate one cross product to determine a triangle face normal. In the SpiderWeb software, we calculate all three cross products and test for agreement of the normal vector signs with respect to each other as a sanity check.

Given each triangle's face normal, we need to know how we can construct smooth vertex normals for the surface. One technique is for each triangle vertex, to average all of the adjacent triangle's face normals. In that way each triangle that shares that vertex will have the same value for that vertex normal. This sharing of vertex normals for all adjacent triangles gives the apparent smooth surface we see in SpiderWeb renderings. The problem this approach creates is in efficiently building adjacency lists. If we wish to explore the surface with some form of steerable surface bug, we need to know exactly which triangle is adjacent to each triangle vertex and edges. We do this by building adjacent lists for each triangle vertex and triangle edge.

Each triangle is shared a triangle edge with one additional triangle. Each triangle vertex is shared with at least 6 additional triangles. Each triangle is owned by a voxel face. We can determine what triangles and their parent voxels are adjacent from the structure imposed by the SpiderWeb algorithm. Consider a voxel face containing a triangle. We can determine all of the adjacent triangles by considering what triangles share the articulating point of the voxel, and what voxels share the



hit edges of the voxel. This makes it possible to rapidly determine triangle adjacency lists.

Having lists of adjacent triangles for each triangle vertex makes it simple to calculate vertex normals from adjacent triangle face normals. The vertex normal is the mean of the adjacent face normal vectors.

An alternate method for calculating vertex normals is to use the volume gradient interpolated to the vertex position. The volume gradient is the vector that represents the direction of steepest descent. This convention is in harmony with the SpiderWeb convention that triangle normal vectors point to the outside, lower density side of the triangle plane. The gradient  $gv$  is defined at a vertex  $v$  by

$$(3.17) \quad gv_i = \frac{v_{i+1} + v_{i-1} - 2v_i}{-2}$$

$$(3.18) \quad gv_j = \frac{v_{j+1} + v_{j-1} - 2v_j}{-2}$$

$$(3.19) \quad gv_k = \frac{v_{k+1} + v_{k-1} - 2v_k}{-2}$$

The volume gradient then interpolated for a subvoxel location is derived from the formula given by Cline *et al.* [3]. The gradient vector  $\vec{g}$  is interpolated only along a voxel edge a fraction  $h$  of the distance between vertices  $v_0$  and  $v_1$ .

$$(3.20) \quad g = gv_0h + gv_1(1 - h)$$

The volume normal can be trilinearly interpolated through the body of the voxel.

We show in Karron *et al.* 1992 [111] that the triangle vertex normals calculated from the volume gradient and from the average of adjacent triangle face normals are not equivalent for triangle vertices on voxel edge hits. For co-planar triangles, the trilinearly interpolated volume normal is not the same as the vertex normals averaged from the mean of the adjacent triangles. This can be intuitively seen from the topological connectivity issues. The volume gradient does not know about topology, and so it will include regions that are topologically disconnected. This distinction becomes important close up in a small scale scene. The advantage of the SpiderWeb normals is that two sheets can pass closely and the surface normals on the nearby disconnected sheets will not influence each other. However, when there is no nearby (less than 1 voxel length away) disconnected surface, the results of the volume normal and the SpiderWeb normal are the same.

### 3.2.3.1 Spinning Triangles in a Voxel Illustrated

The principal operation of the SpiderWeb algorithm is spinning triangles. Essentially, we traverse each of the connected faces in a voxel, starting from the voxel entry face, and the entry voxel edge, and cycle (spin) triangles from the connected hits on the connected edges in the connected faces with the connected articulating point.

A SpiderWeb surface is tiled from triangles that have a well defined relationship to the parent voxel. Each triangle has a triangle vertex that is congruent with the articulating point. This vertex is the **triangle articulating vertex**. The other two triangle vertices are congruent with the voxel edge hits. These triangle vertices are referred to as **hit vertices**. One triangle edge is composed from the two hit vertices that lie on, and cross the voxel face. This edge is called the **triangle voxel face edge**. The two remaining triangle edges are the **emerging triangle edge** and **plunging triangle edge**. This is because the triangle edges are knit by crossing the surface of the voxel face, then plunging into the body of the voxel to the articulating point, then emerging back to the surface of the voxel face. Because SpiderWeb surfaces are manifold, each triangle edge must be resolved, or be adjacent to, one more triangle. An **unresolved edge** is a triangle edge without a continuing triangle adjacent to it.

We now show that within a voxel, there are no un-resolved triangle edges that do not lie on voxel faces. We showed in 3.2.2.2 that a triangle tiled surface will not have holes if it is voxel face-wise continuous. We can consider each voxel face independently for the purpose of spinning triangles. However, each voxel face is not independent because hits propagate to other, adjacent voxel faces. A pair of hits in a voxel face causes at least two more voxel faces to have hits. A single triangle is generated for each connected voxel face with connected hits. This separates two triangles knit in a 4 hit face, or bubble triangles in SpiderWeb A. Each triangle in a connected system of hits shares the articulating point. Each triangle in the connected system also shares one voxel edge hit between two triangles. Therefore, this shared hit and the shared articulating point create a shared edge. The other hit (recall, hits must come in pairs) is also shared by two triangles. Because all triangles in a connected group share the articulating point, every triangle edge that is not on a voxel face is shared by two triangles, and there are no possible gaps or holes in the triangle surface within voxels.

Since we have shown that if the voxel face-wise continuity rule is observed, the last remaining triangle edge is shared. This proves that every triangle in the

SpiderWeb surface is manifold because each triangle edge is shared by two triangles. Equivalently, we can say that every triangle edge is resolved by another triangle. There are no unresolved edges, and the surface is complete, without any holes.

We now demonstrate the construction of triangles with the SpiderWeb algorithm. Our first example is to build a simple set of three triangles in the corner of a voxel.

Consider a voxel with all vertices except for one in the Out state. The analysis of triangle spinning can proceed from the point of view of the voxel face. This results in considerable simplification in analyzing other vertex states.

Figure 3.28 shows the resultant pattern of hits on the voxel face caused by one hit. We cycle through the hits by spinning a triangle from the first voxel edge hit across the voxel face to the second voxel edge. This satisfies the voxel face-wise continuity condition. We continue through the cycle by spinning a triangle edge from the second voxel edge hit inward to the articulating point. We complete the cycle by emerging from the articulating point to the first voxel edge hit point. We have completed spinning the triangle in this voxel face.



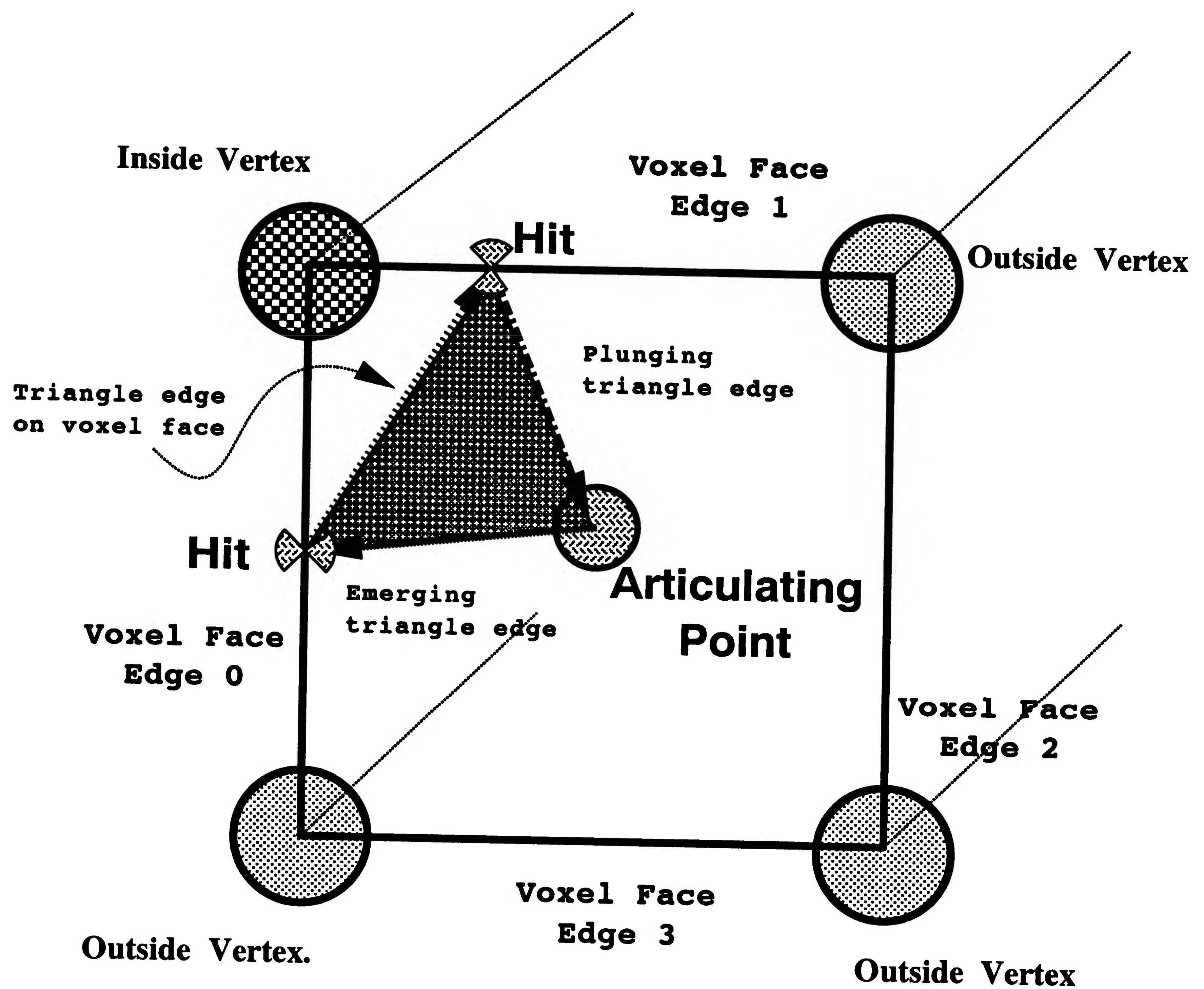


Figure 3.28: Spinning a single triangle in a voxel face.

We now look at the connected voxel faces produced by the system of hits. In Figure 3.29 we see the assemblage of three triangles produced by the three voxel faces containing hits. Each face is examined independently, and as we have shown above, a complete surface within the bounds of the voxel is generated.

This triangularization “slices off” the corner of a voxel, and is called a **voxel corner triangle**. The three co-planar triangles can be easily reduced to 1 triangle if desired. The main reason not to pluck the articulating vertex to generate a single triangle is that it breaks up regular triangle mesh runs. More information about issues in triangle meshing co-planar triangles is discussed in Section 3.2.3.4. Additionally, as we will see in the next sections, the surface constructed in this way will be maximally smooth within the voxel. Removing triangles will coarsen the surface. SpiderWeb surfaces will have the smooth crease lines.

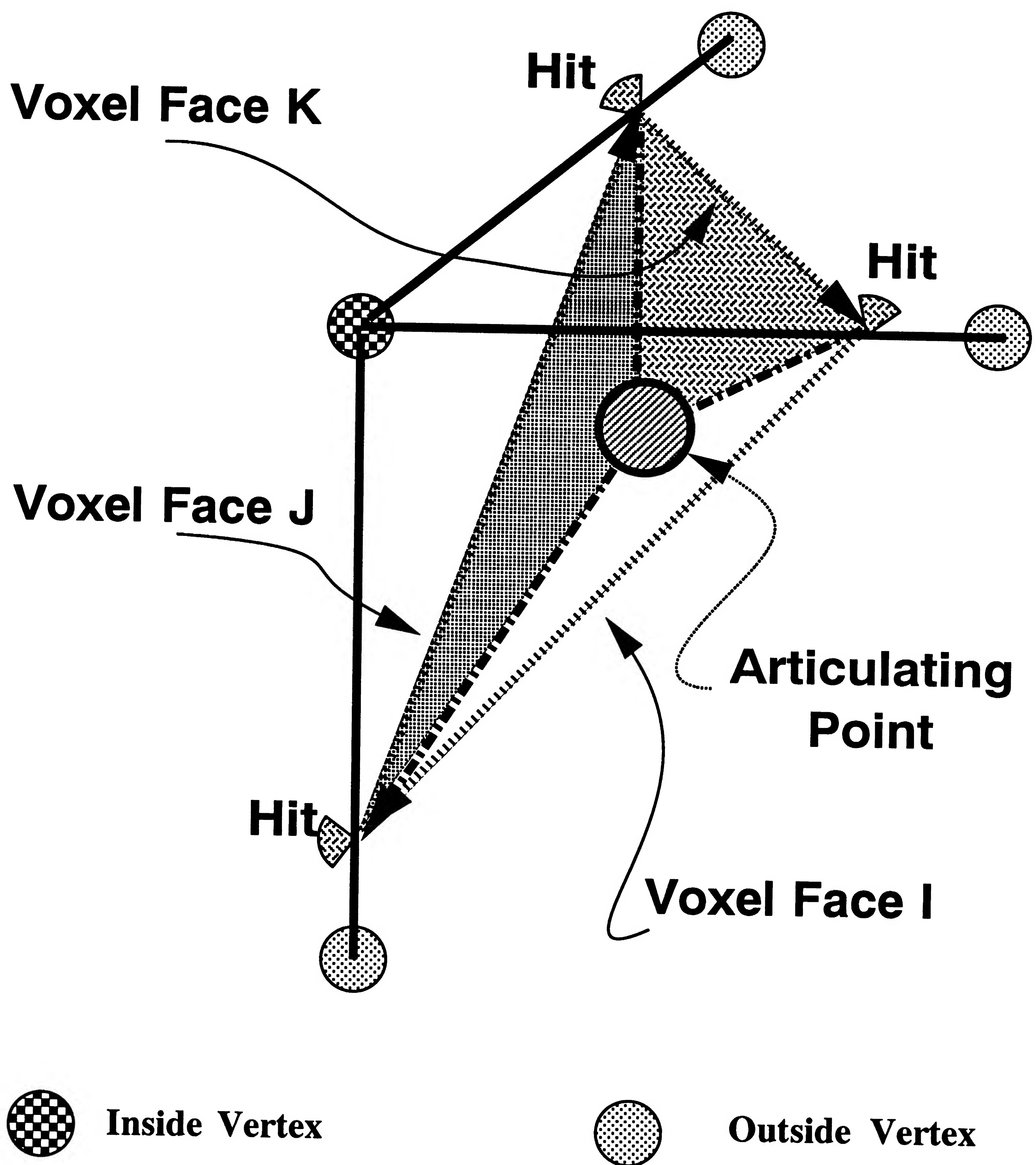


Figure 3.29: Three triangles in three connected faces spins a locally connected surface.



In this example, we look at a configuration of voxel hits about the “waist” of a voxel. Such a system of triangles bisects a voxel about the middle, and is a system of “waist triangles”.

A few other combinations with two hits on each voxel face are possible. The construction and analysis is exactly the same as given already. There is no need to enumerate all cases, as we have proved that no matter what particular situation we encounter, we will generate a legitimate surface. A proof by enumeration of the SpiderWeb algorithm is not required as it is for Marching Cubes. Indeed, I believe that it should be quite simple to compile a voxel case lookup table with the SpiderWeb algorithm that will be quite similar, if not exactly the same, as Lorensen and Cline’s Marching Cubes case table. The main strength of the SpiderWeb is that we can justify each knitting decision and choice as a generalized triangle spinning rule.

If there are only two hits on all of the voxel faces, then the geometry is completely unambiguous. This was shown for two dimensional boxels in Section 3.1.1.4.

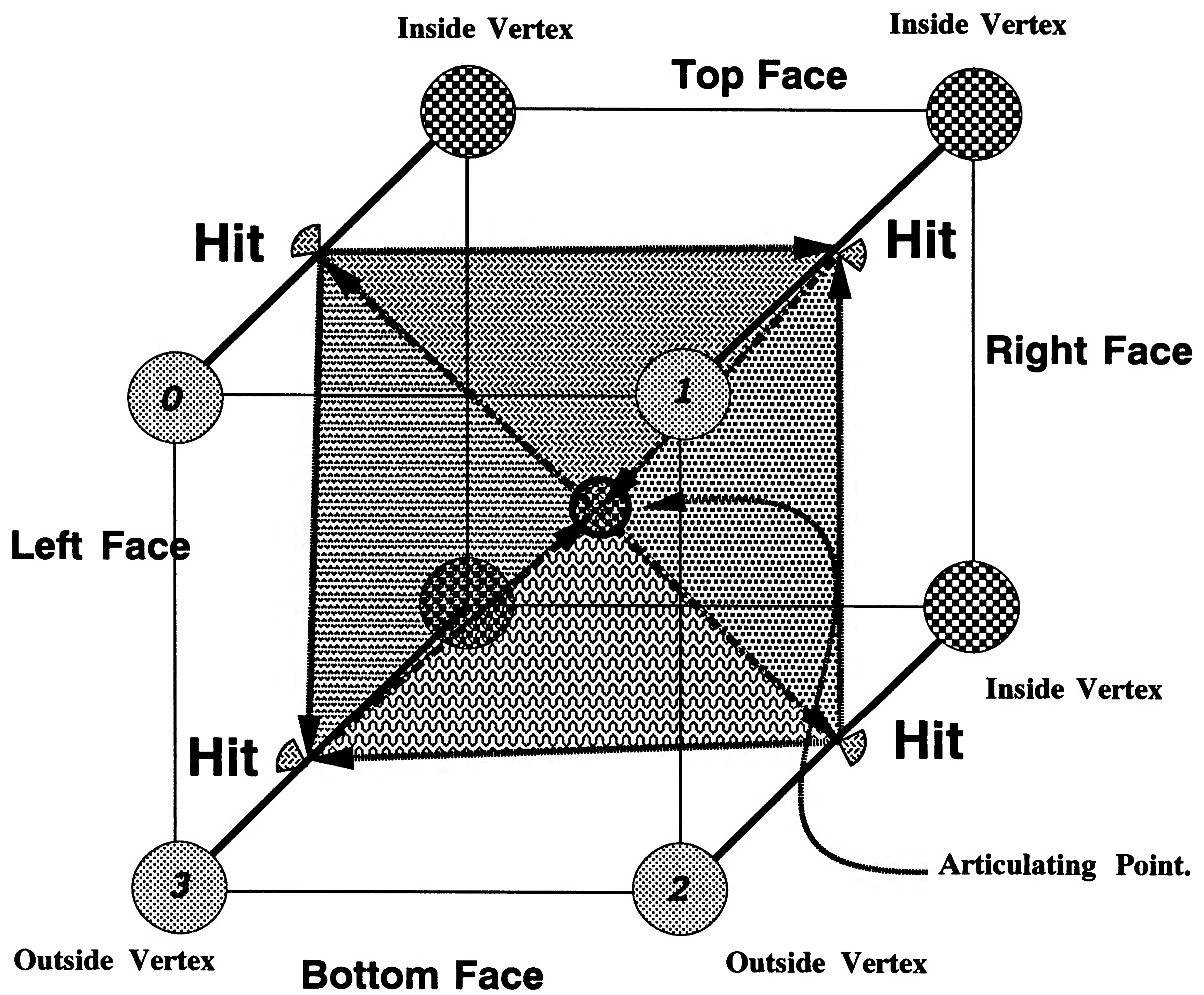


Figure 3.30: Four triangles in four connected faces spins a connected surface.

We have already seen in Section 3.1.1.5 that three hits on a face is impossible.

The next class of SpiderWeb boxel hits involves knitting choices for voxel faces with 4 hits per face. The development of the options for 4 hits per face voxels is almost the same as for boxels that was given in Section 3.1.1.6. The essential difference is that we have more information provided in three dimensions to decide which way to knit surfaces for the SpiderWeb B algorithm. The SpiderWeb A algorithm does not make a decision, and spins a bubble triangle as shown in Figure 3.31.

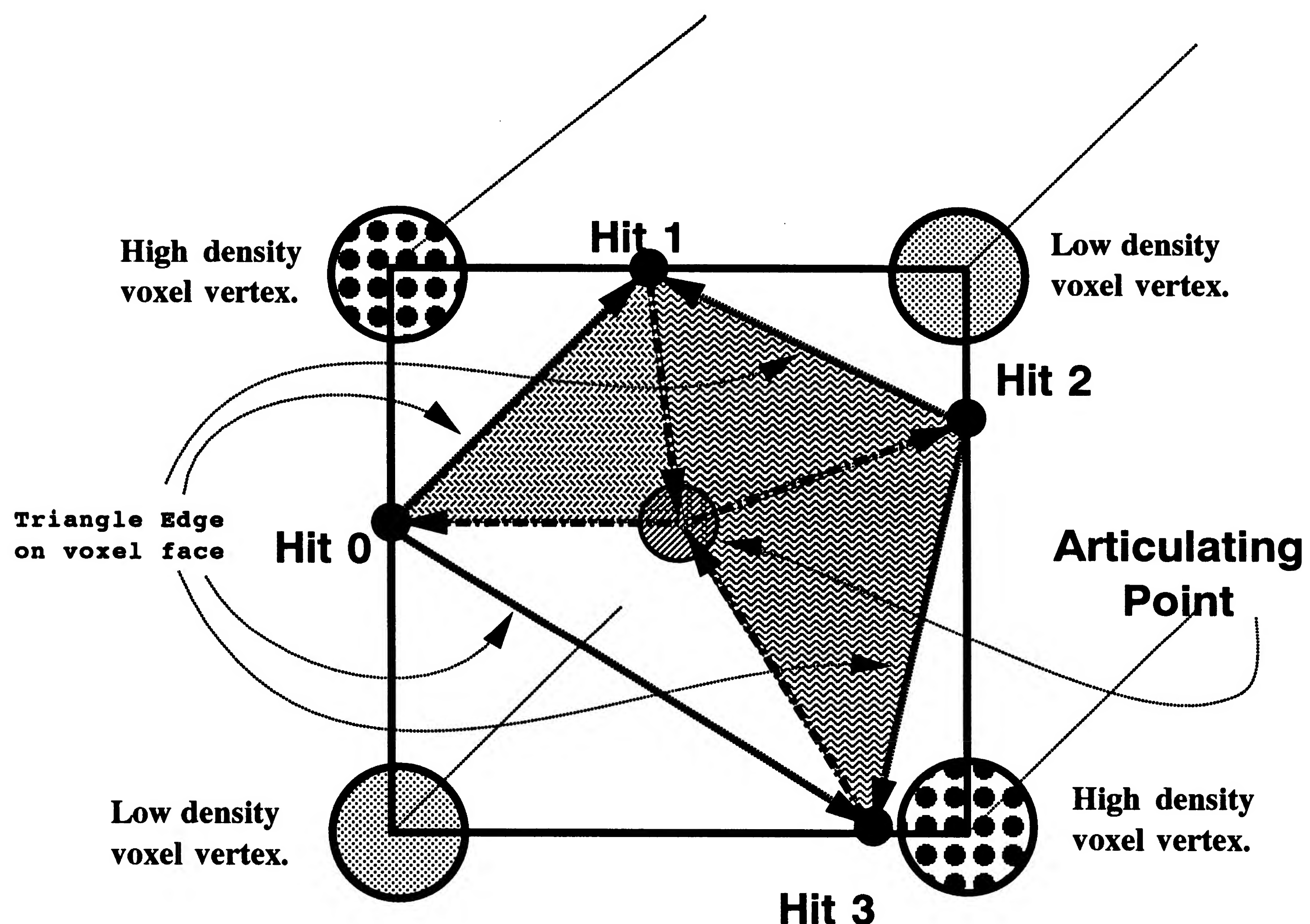


Figure 3.31: Four triangles in one faces spins a bubble in the surface for SpiderWeb A.



We have examined the assembly of individual triangles. We now will see how these triangles assemble into large scale geometrical representations of anatomic features. Figure 3.32 shows a single voxel from the SpiderWeb program. The small figures at the vertex of the larger voxel represent inside vertices. The triangles are drawn inside the voxel as wireframe figures. The triangle vertices are congruent with the voxel edge hits. This particular voxel has hits in 5 faces.

The same voxel in Figure 3.32 is shown with the triangle face normals drawn in “hedgehog mode”. A larger group of voxels is shown with normals in Figure 3.34. The size of each voxel edge is 1 mm. We are seeing the bottom of a sharp bend in the brain surface geometry. As we pull away from the group of voxels in Figure 3.32, we see in Figure 3.35, a much larger group of triangles, almost 30,000, that tile the walls of a brain sulcus. Each triangle is about 1/2 mm in size. The width of the sulcus is about 3 mm wide. Figure 3.36 shows the same triangles in hedgehog mode. Figure 3.37 shows a close up view of the entrance to this sulcus. The triangles are again approximately 1/2 mm in size. Figure 3.38 is a far view of the same sulcus in Figure 3.35. This view shows the orientation of the sulcus with respect to the entire head. The axis partly visible behind the slice shows the approximate orientation of the head-based coordinate system, with the  $-X$  axis oriented toward the occipital pole of the head. Example skin surface reconstructions are shown in Section 6.0.2.6.

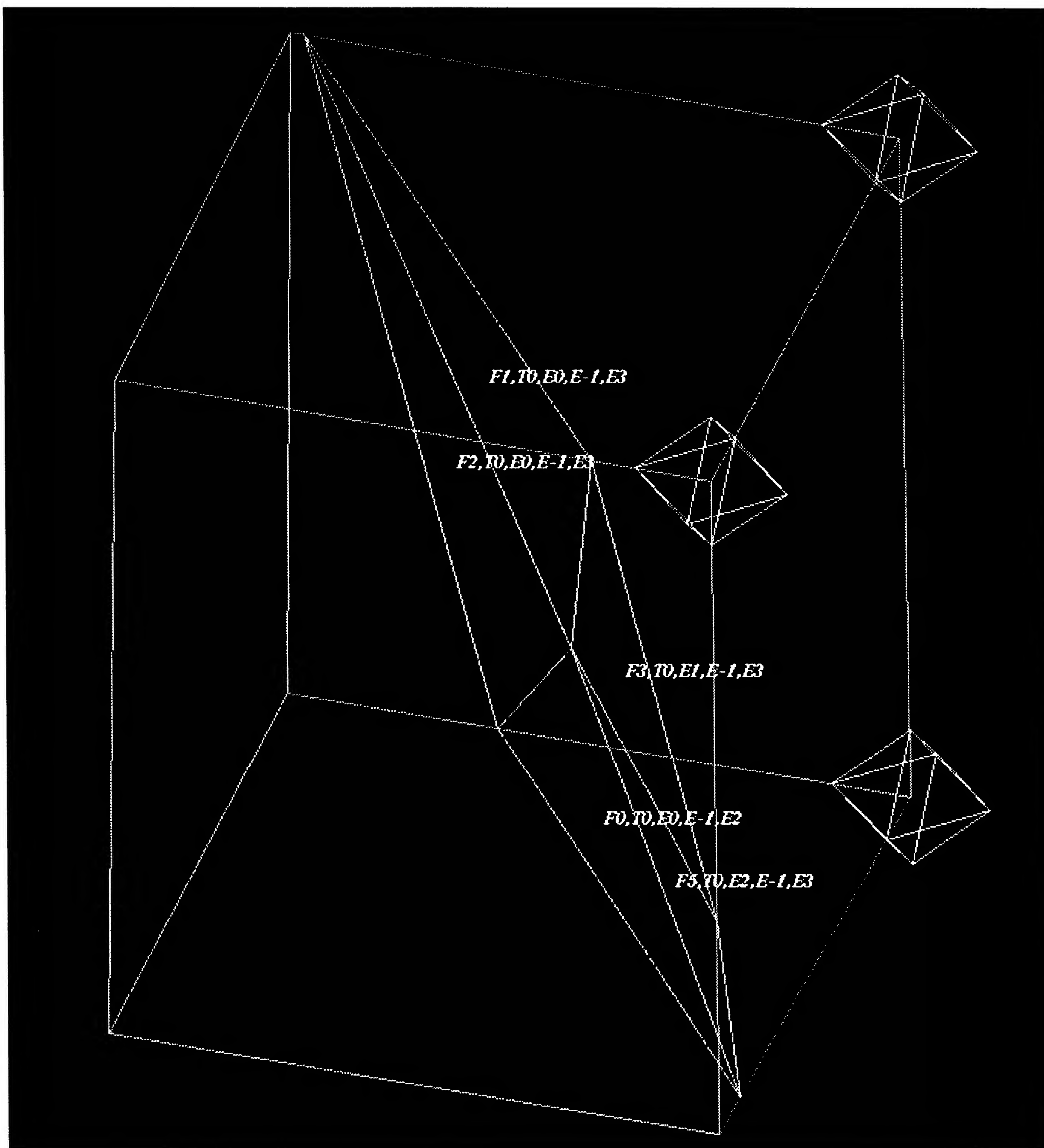


Figure 3.32: An example triangle webbing inside a voxel.

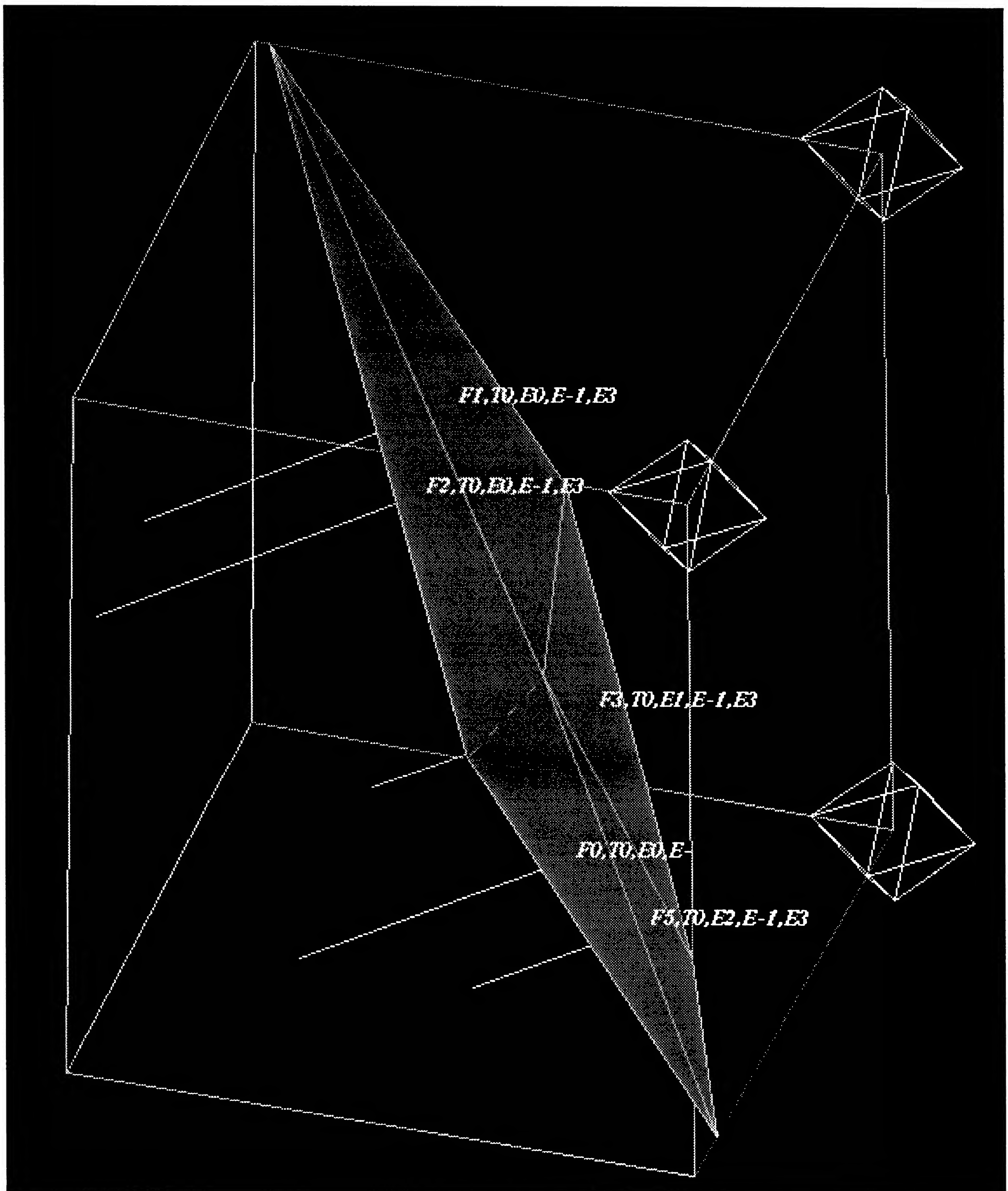


Figure 3.33: An example triangle webbing showing triangle normals inside a voxel.



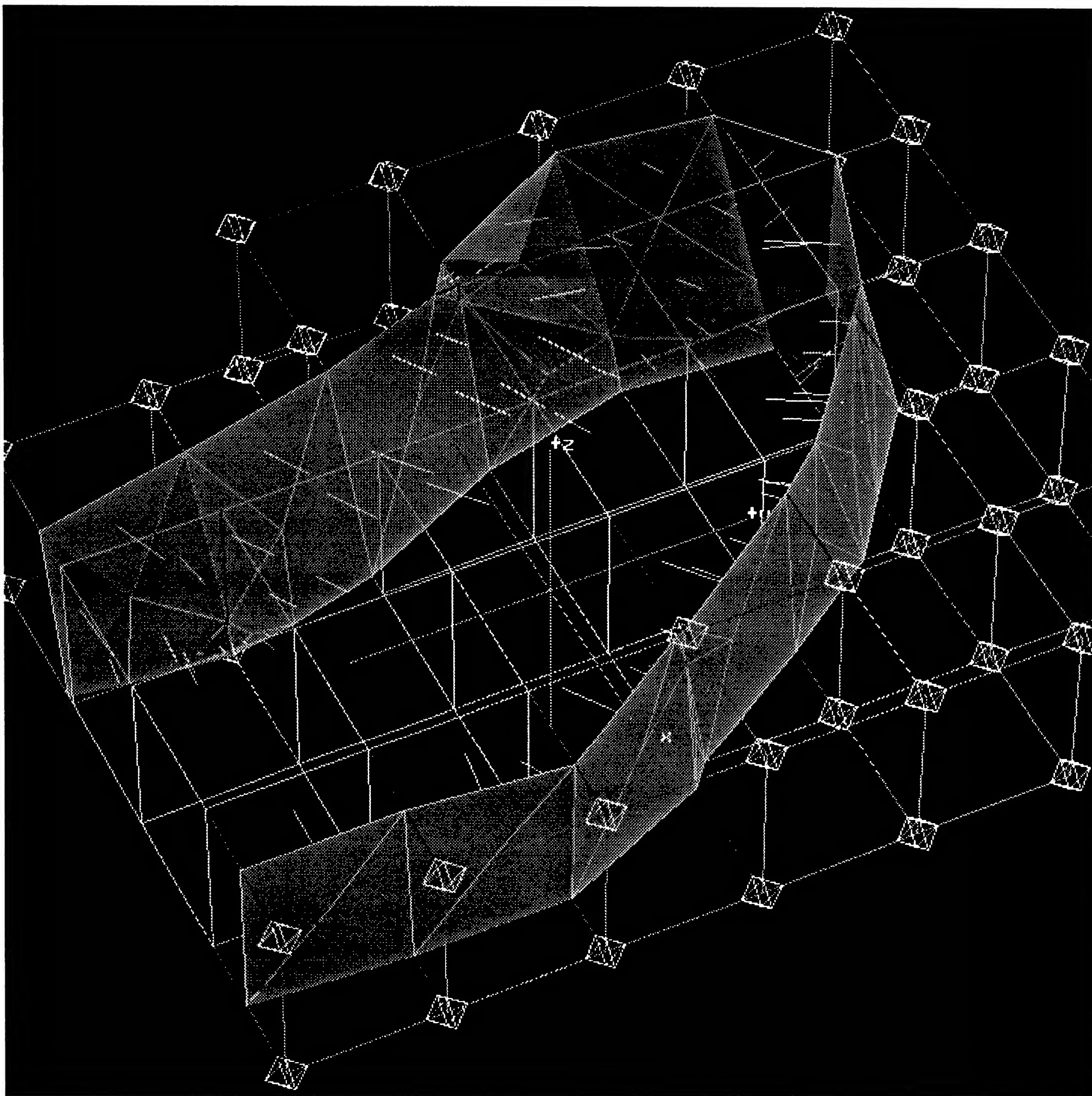


Figure 3.34: A SpiderWeb triangles assembled in a small group of voxels.



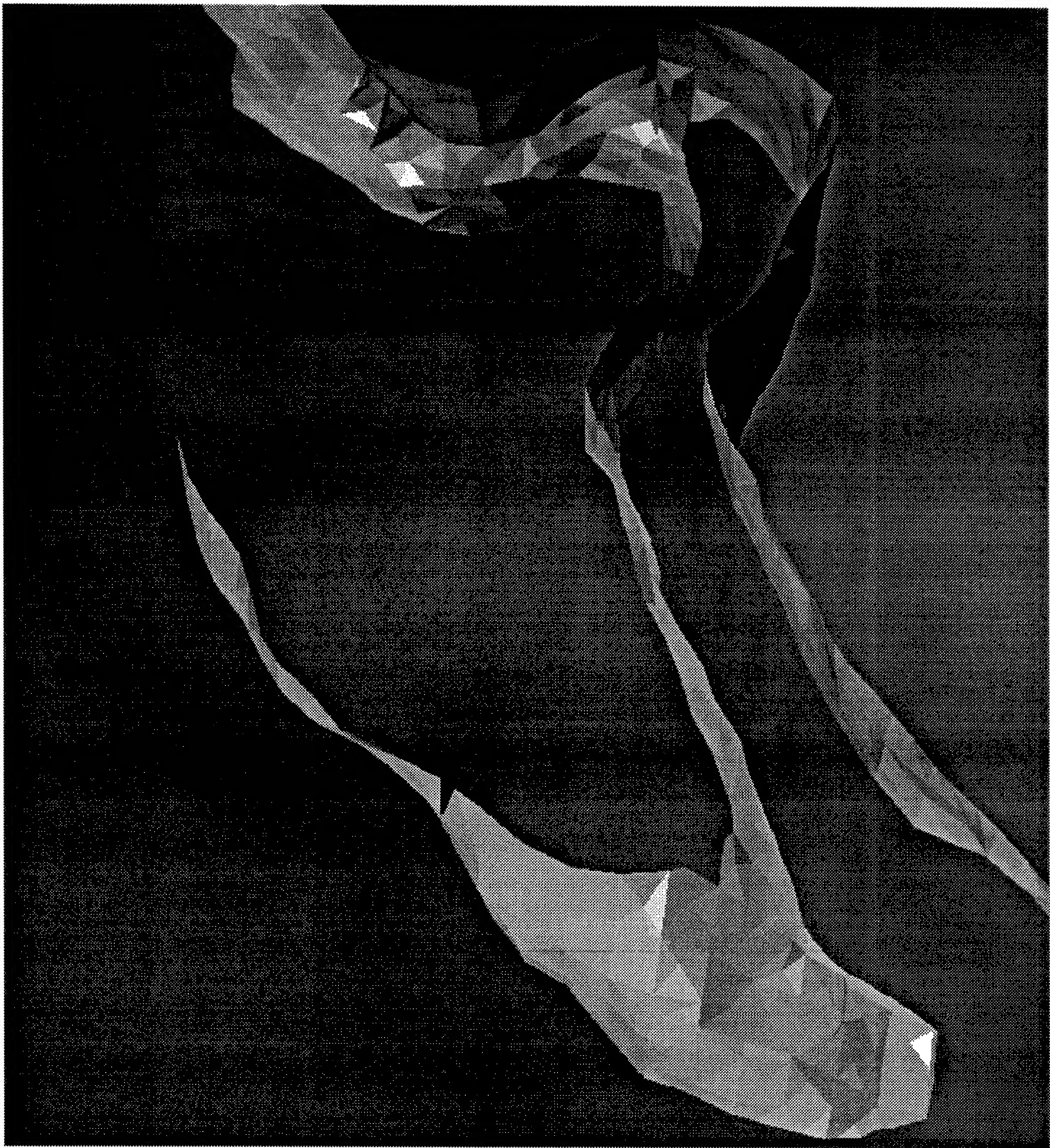


Figure 3.35: An example brain sulcus surface, extending deep inside the head. The fissure is approximately 3 pixels (3 mm) wide.



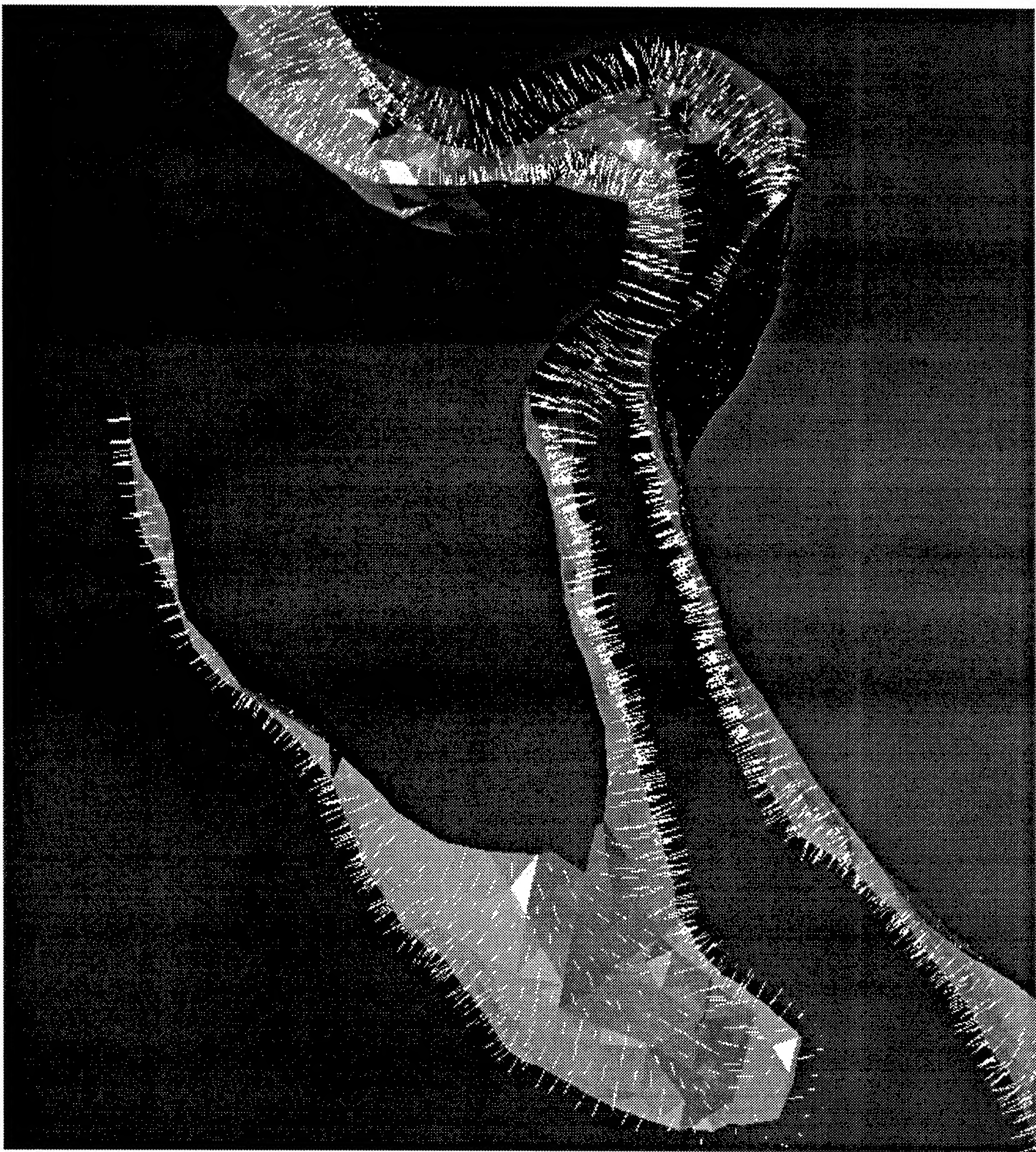


Figure 3.36: An example brain sulcus surface, showing triangle face normal vectors(“hedgehog mode”).



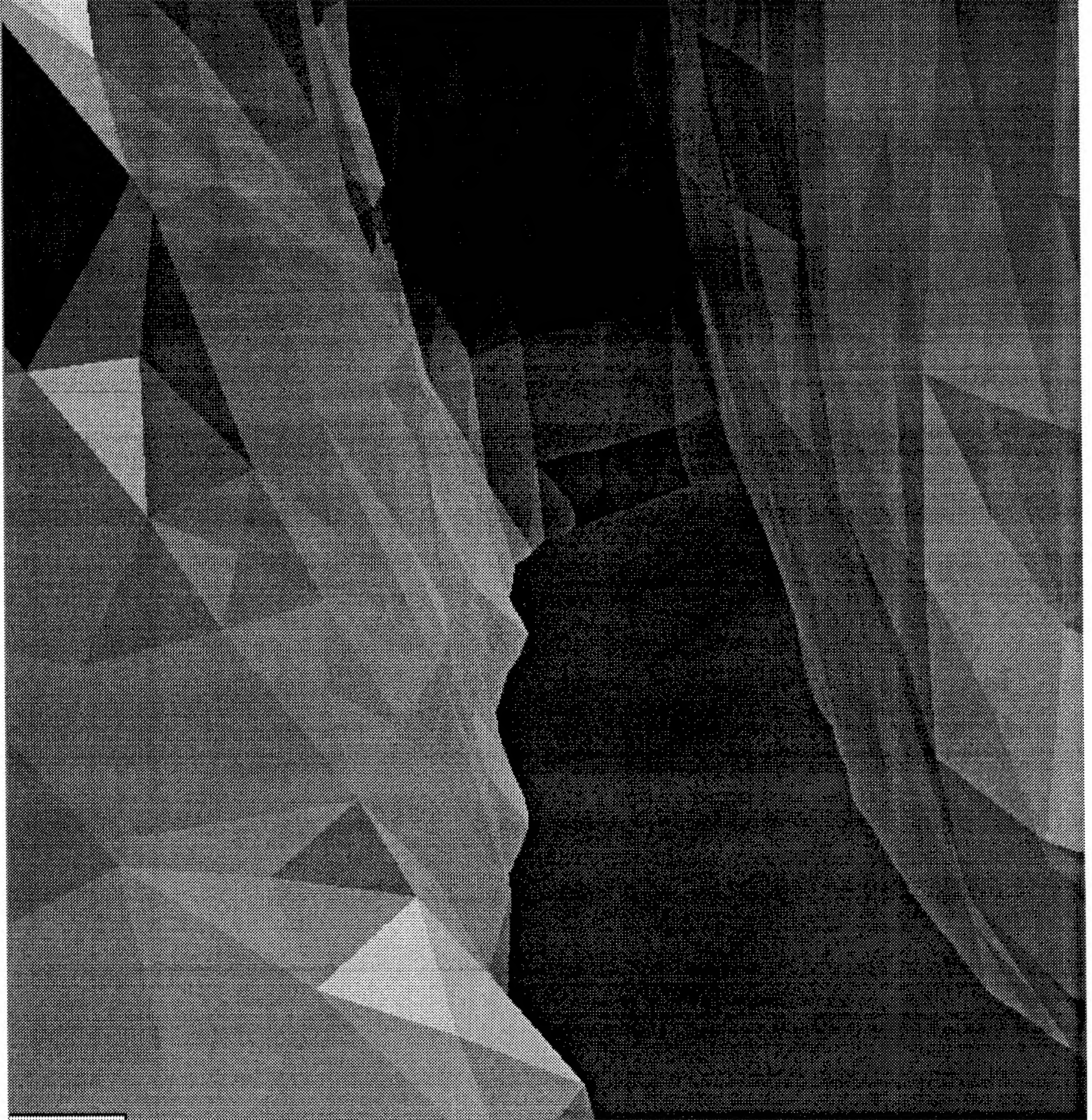


Figure 3.37: A close up view of the entrance to the sulcus.





Figure 3.38: A far view of the partial surface reconstruction of a sulcus extending below an axial MRI slice. The width of the fissure is 3 mm, the size of each pixel is 1 mm.



### 3.2.3.2 Comparing Volume and Surface Rendering Algorithms: Volume Occupancy and Depth Complexity

There are a number of issues in determining the relative merits of volume rendering *vs* surface construction. Two ratios can help evaluate the ultimate maximum drawing speed and complexity issues when comparing volume *vs* surface algorithms. As the data lattice is traversed during drawing, each part of the lattice that requires drawing is transformed and projected onto a windowed region of the viewers screen. Different algorithms project the object being drawn in the data to the screen in different ways. We can compare the relative efficiencies of different algorithms by two theoretical ratios. They are Volume Occupancy and Depth Complexity. All references to volume, surface, and voxels are illustrated with area, contour lines, and boxels in Figure 3.39, but the concepts generalize directly to 3D. Voxels can be used in the pixel-point sense (voxel is a box around a point sample) or the pixel-spanning sense (voxel is a box between 8 pixel vertices).<sup>1</sup>

Volume occupancy (VO) is the fraction of the voxels that contain a bit of the surface under consideration.  $N_{VoxelsOccupied}$  is the number of voxels that contain a surface triangles.  $N_{TotalVoxels}$  is the total number of voxels in the scene. The VO is

$$(3.21) \quad VO = \frac{N_{VoxelsOccupied}}{N_{TotalVoxels}}$$

Figure 3.39 lattice A shows a simple contour. In this scene, or region of interest, a volume occupancy of 10/25 means that the surface does not occupy a large fraction the region. An extremely convoluted surface such as shown in Figure 3.39 lattice B, has a Volume Occupancy value approaching 1.0, meaning the surface passes through almost every (boxel) voxel in the (area) volume.

The depth complexity (DC) is the average number of times that each particular pixel is drawn in rendering a particular volume region of interest.  $N_{WindowPixels}$  is the number of pixels drawn in a particular window.  $N_{PixelDraws}$  is the number of times on average, that each pixel is drawn in rendering a scene in a window.

$$(3.22) \quad DC = \frac{N_{WindowPixels}}{N_{PixelDraws}}$$

A small ( $\ll 1.0$ ) Depth Complexity means that each pixel in a screen window is drawn once. This is characteristic DC of drawing a flat plane, 2D objects (that do

---

<sup>1</sup>SpiderWeb voxels are always pixel-spanning.



not overlap or are co-incident). Any closed 3D object, such as a SpiderWeb surface shown in Figure 3.39 lattice A, in that is grown to completion, must have a Depth Complexity  $> 2.0/N_{WindowPixels}$  for any region of interest that encloses the object. This is because any front surface must have a back surface, and any ray passing through this surface must intersect the surface at least twice. A highly convoluted object, such as a very noisy surface will have a depth complexity approaching 1.0, as can be seen in Figure 3.39 lattice B.

This property can be used to differentiate from Inside pixels and Outside pixels. Consider the first time we draw a bit of the surface in Figure 3.39 lattice A. What is displayed on the screen is the inside of the volume enclosed by the surface. This is an inside pixel. The next time that pixel is drawn, it represents the outside of the surface. This is an outside pixel. In a front to back data traversal, inside pixels are drawn first, and outside pixels are drawn second. This geometric property is used in Silicon Graphics VGX computer hardware stencil buffers to track front and back, or inside and outside surfaces when rendering cutaway polygons [126]

An extremely complicated object, with intricate convolutions and folds will have a large Depth Complexity. A simple closed 3D object will have a Depth Complexity of  $2/n$ . A 2D object that is not closed in 3D space will have a Depth Complexity of  $1/n$ .

These two numbers can be used to evaluate the ultimate complexity of drawing a three dimensional scene from a three dimensional data lattice. Actual computation of the VO and DC are not generally useful or practical. These figures provide a theoretical basis for comparing a recursive surface tracking approach versus an iterative Lorensen and Cline [62] Marching Cubes or a Vital Images [127] volume rendering approach.

Volume rendering algorithms generally traverse the entire volume of data for each draw cycle. For a  $256 \times 256 \times 256$  data array, the effective volume occupancy is  $1.0/n$  and the depth complexity is  $256/256$ . The entire volume must be traversed, even if there is no geometric significance. the depth complexity is  $256/256$  because each slice of the volume must be drawn to the screen, and the same pixels are redrawn depending on their opacity. Newer volume rendering algorithms that build trees of linked voxel lists that contain non-transparent voxels reduce the volume occupancy and depth complexity to values that approach surface rendering algorithms, but do not generally produce a surface geometry.

In the case a noisy or intricate surface in a region of interest, where the DC is large and the VO approaches  $1.0/n$ , there is no theoretical advantage of the SpiderWeb or other recursive surface tracking algorithms. The main problem with this class of volumes is that the interior details are obscured by exterior features. Various cutting and segmentation planes are used to expose the interior detail. Recursion overhead and the number of bubble voxels increases as the radius of curvature of the intricate or noise decreases, iterative approaches will be more efficient in these cases.

The main advantage of surface rendering is for simple geometries with small VO and DC fractions. Consider that any object of topological genus 1, such as a sphere, will have a DC of  $2/n$ , and a VO approximately of  $2/n$ . Intuitively, a sphere has a front and a back surface, and that at least two voxels must be occupied for any ray that maps a column of voxels into a screen or window pixel. For relatively simple surface objects embedded in a volume, characterized by relatively small VO and DC, a recursive octree or surface tracking algorithm has comparatively greater speed.

Voxels can be preselected when they contain density information within a certain threshold range. These voxels can be tagged, and a tree constructed giving connectivity information. These voxels can be recursively visited and have many properties similar to a surface tracking algorithm. This approach was suggested by Bloomenthal[72][73]. An octree voxel list will have the same VO and DC as a the SpiderWeb algorithm, is discussed by Wilhelms and Van Gelder[128]. The main advantage of the SpiderWeb algorithm to octree approaches is that by geometric continuity, we never have to probe adjacent voxels to check for occupancy. We know before hand which way the surface extends to, and we never step into an empty voxel. We can expect convergence of two technologies now that the underlying geometry of any possible surface construction is understood with the SpiderWeb algorithm.

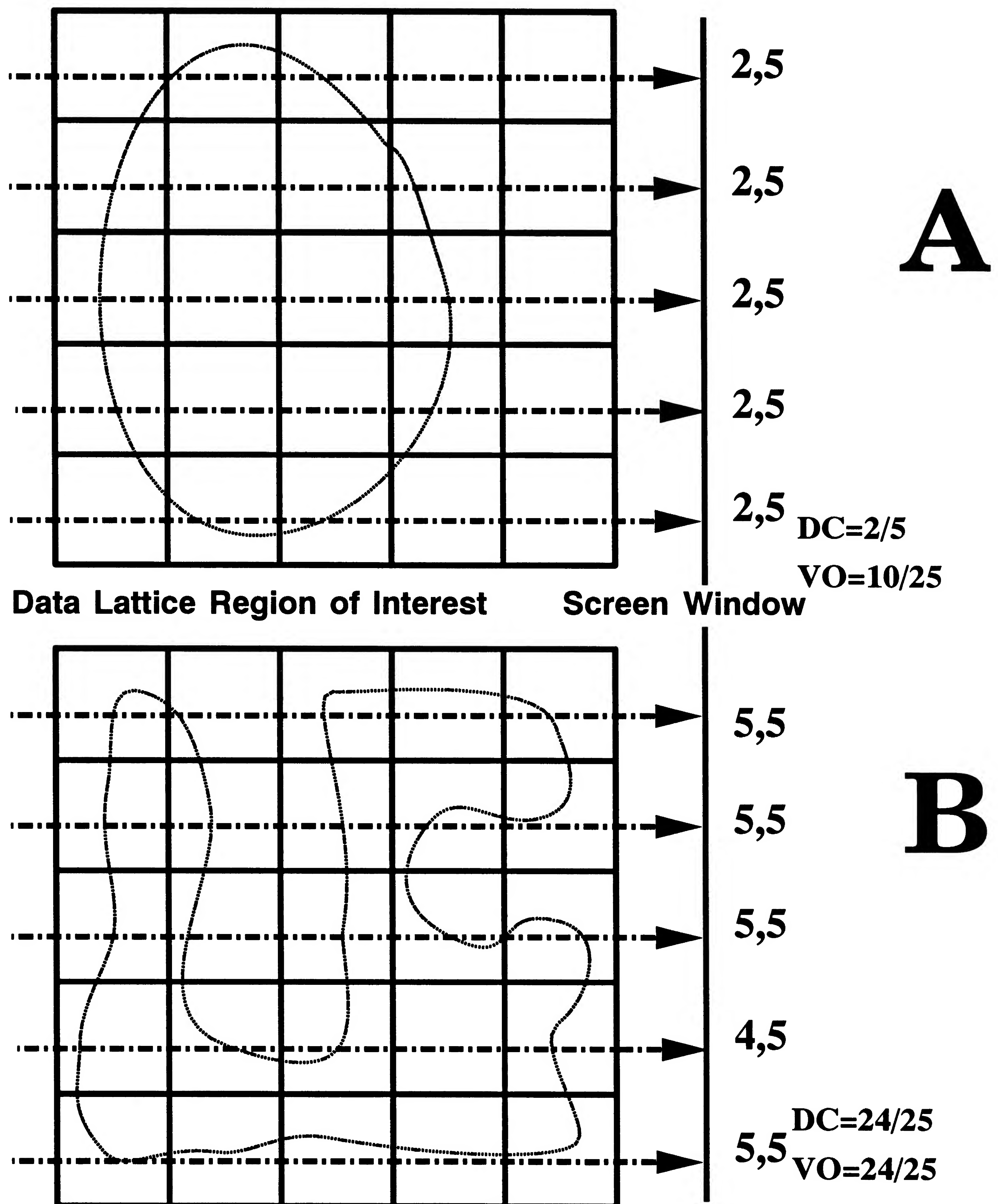


Figure 3.39: Depth complexity and volume occupancy in a region of a data lattice.



### 3.2.3.3 Simplifying SpiderWeb geometry

A major problem with the SpiderWeb algorithm is that it can create an excessive number of triangles. An excessive triangle contributes very little to the essential geometry of the object being constructed. For example, a connected system of three hits on a voxel corner, generates three co-planar triangles, as is shown in Figure 3.29. The center, articulating vertex can be immediately plucked out to yield a single triangle. For decreasing voxel size, and they become more numerous, and the high resolution surfaces produced by the SpiderWeb algorithm are of less value. We would like to reduce the geometric detail, while preserving the essential topology.

If an edge has a hit, we can move the hit to a standard position midway between vertices instead of lineally interpolating a position based on density along edges. The advantage of this is that the angles between surface triangles is reduced to a small number, effectively making the surface look blocky in detail. The degree of blockyness, or the blockyness factor  $B_f$ , can be varied from 0.0, for no hit drift to the center to 1.0, in which case all hits to shift to the midpoint of the edges. Explicitly, the formula for blocky hits is:

$$(3.23) \quad E_{hx} = V_{x0} + (V_{x1} - V_{x0}) \left( \frac{\tau - V_0}{V_1 - V_0} \times (1 - B_f) + \frac{B_f}{2} \right)$$

$$(3.24) \quad E_{hy} = V_{y0} + (V_{y1} - V_{y0}) \left( \frac{\tau - V_0}{V_1 - V_0} \times (1 - B_f) + \frac{B_f}{2} \right)$$

$$(3.25) \quad E_{hz} = V_{z0} + (V_{z1} - V_{z0}) \left( \frac{\tau - V_0}{V_1 - V_0} \times (1 - B_f) + \frac{B_f}{2} \right)$$

Figure 3.40 shows the effect of increasing  $B_f$  on a hit on an edge near a vertex. As  $B_f$  increases, the hit drifts to the midpoint of the edges. The apparent corrugation in the smooth surface is mainly as a result of the slice orientation “grain”. Figure 3.41 shows a bit of forehead surface that has been rendered with  $B_f = 1.0$ . Figure 3.42 shows the same surface rendered with vertex normals, which give a much smoother rendition of the surface. Figure 3.43 shows the results of an intermediate  $B_f = 0.5$  with triangle face normals, and Figure 3.43 shows the smoothed rendition with triangle vertex normals. The main advantage of this technique is that the underlying geometry can be made quite simple, and a large number of co-planar triangles can be removed to speed drawing. The essential topological relations between the surfaces is preserved despite the coarsening of the surface.

If we have a dataset where each pixel has been segmented into some binary state, such as bone and not-bone, an interesting situation arises. Such a segmentation is calculated by assigning a value of 1 to bone, and a value of 0 to not bone. In such a lattice where all vertices have a binary density value, we see the same blocky surfaces we obtain by shifting hits here. In a binary volume data, all hits occur at the same relative position along edges. The particular threshold  $\tau$  does not change the geometry. Indeed, we would choose  $\tau = 1/2$  to obtain all hits precisely at the midpoint of all hit edges. This implies that a lot of subtle, possibly important geometric information is lost in binary segmentation. The geometry of such a system contains blockyness artifacts. There is a lot of research on geometry and segmentation issues I hope to undertake to understand the meaning of these algorithms, both in computational geometry and medical visualization practice.

Isosurface Threshold=59.0

Blockyness = 0.0



$V_{i,j} = 1.0$

$V_{hit} = 59.0$

$V_{i+1,j} = 60.0$

$X = 0.0$

$X = 0.9$

$X = 1.0$

Blockyness= 0.5



Blockyness= 1.0

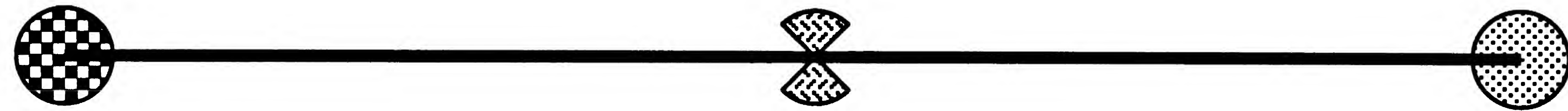


Figure 3.40: Blockyness Factor shifts hits to the midpoint of all edges.



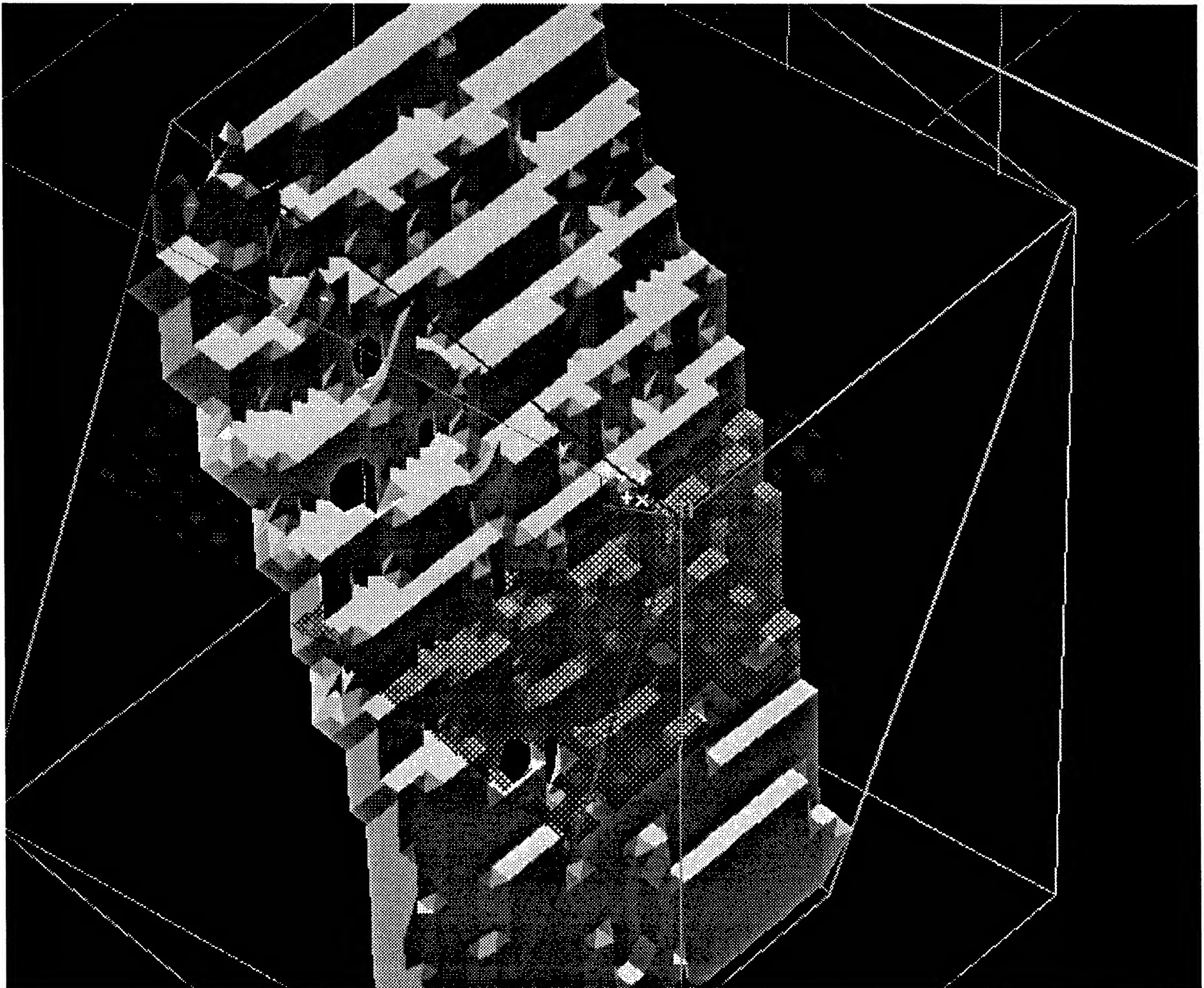


Figure 3.41: Blockyness factor set to 1.0 with triangle face normals used.



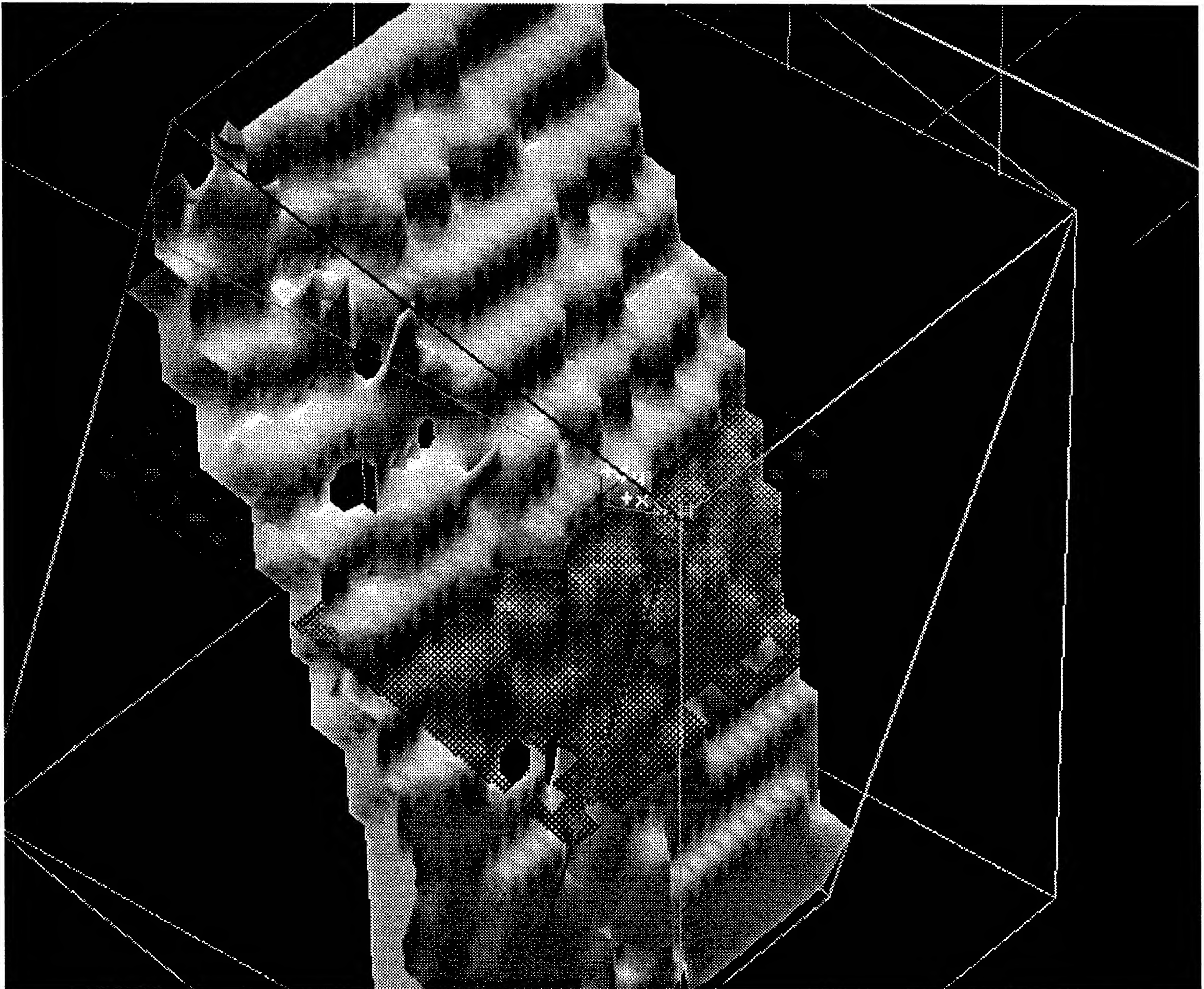


Figure 3.42: Blockness factor set to 1.0, with triangle vertex normals used.



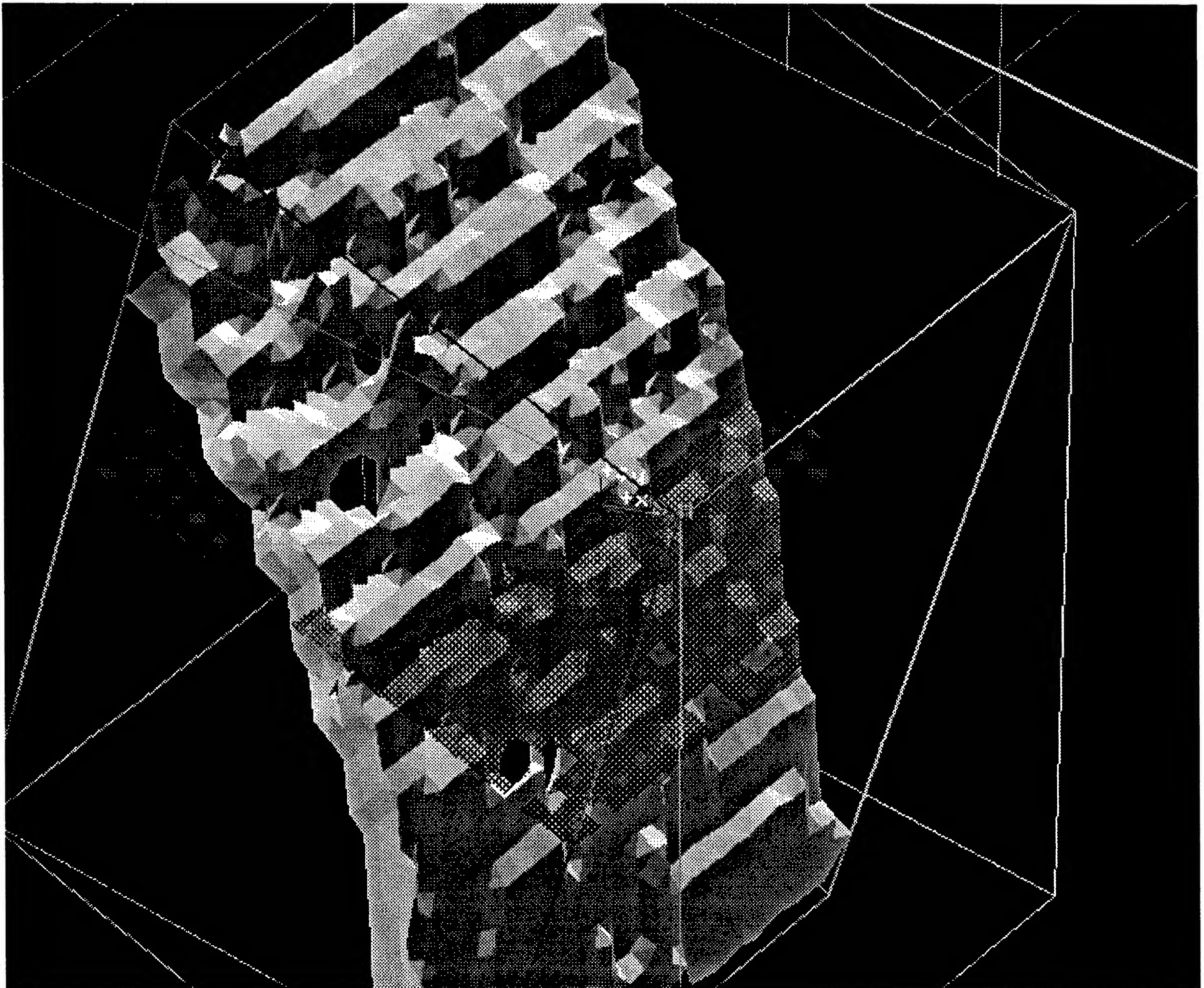


Figure 3.43: Blockyness factor set to 0.5, with triangle face normals.



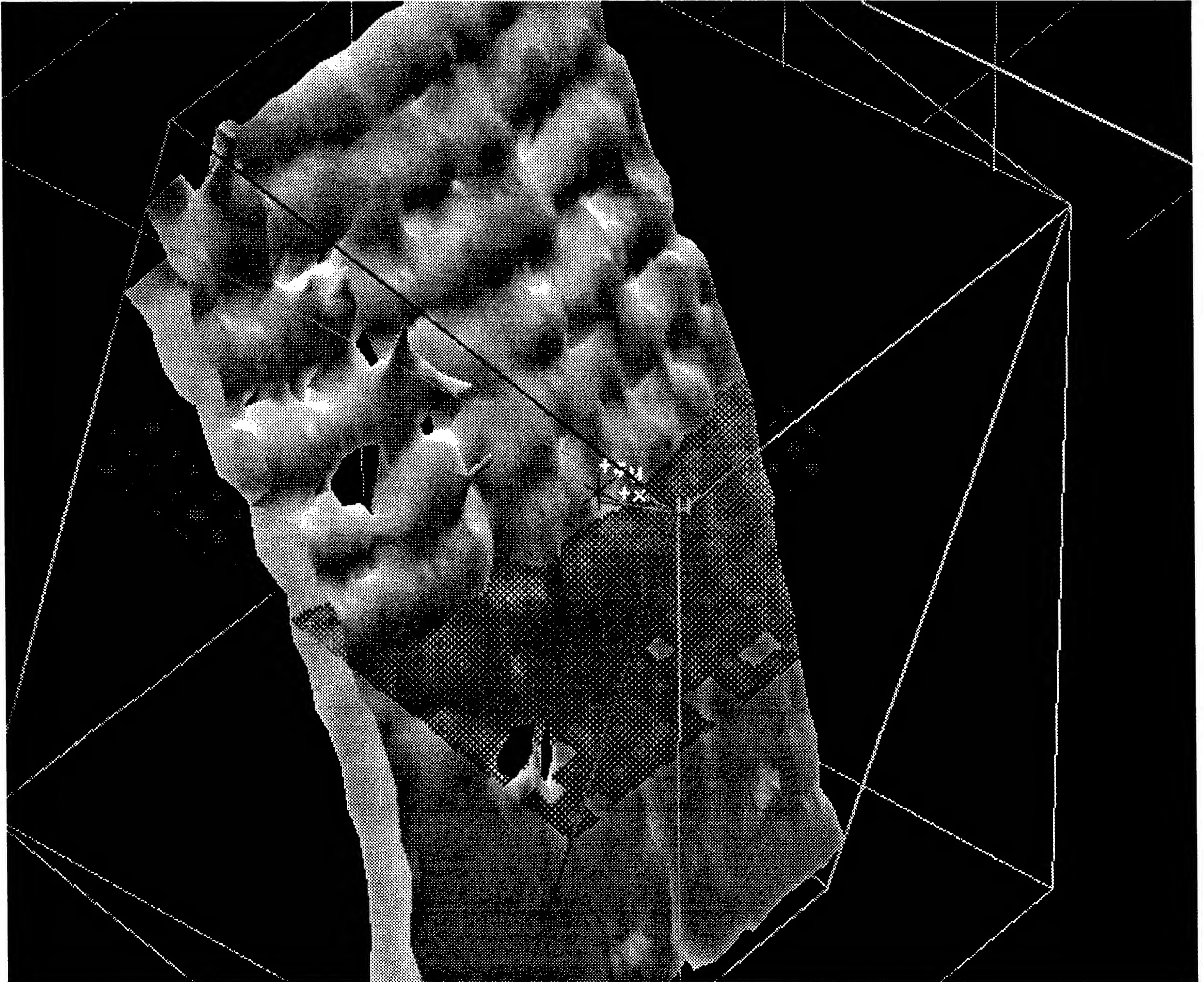


Figure 3.44: Blockyness factor set to 0.5, with triangle vertex normals.



#### 3.2.3.4 Fast Triangle Mesh Generation From SpiderWeb Tiling

A T-mesh run is a contiguous strip of T-mesh triangles, as shown in Figure 3.45. The “zig-zag” ordering of vertices is referred to as T-mesh order. The idea of a T-mesh may be that we specify an initial three triangle vertices, 0, 1, 2. After vertex 2 is specified, triangle 0 is drawn, using vertices 0, 1, 2. After vertex 3 is specified, triangle 1 is drawn, using vertices 1, 2, 3. Vertex 4 builds triangle 2, and so on. After the first three vertices, a complete triangle is drawn for one, instead of three, vertices. This is because once the T-mesh strip is started with the first triangle given by three vertices, each new vertex defines an entire triangle. Each triangle is built on the previous one, and there is no need to redraw shared triangle edges and vertices that are already in the framebuffer.

We can build up a triangle mesh by traversing voxel faces and a row of triangles as a group. Triangles in this configuration can be drawn quite fast. Silicon Graphics hardware can achieve an ultimate triangle drawing rate of 1,000,000 triangles per second. The main problem in achieving this tremendous throughput is finding a fast and efficient means to take triangles produced via other algorithms and ordering them in triangle mesh order. Hansen’s parallelized Marching Cubes algorithm generates 170,000 triangles per second. The Silicon Graphics VGX hardware could draw about 150,000 non T-meshed triangles per second, as clocked by Hansen. Clearly, triangle mesh ordering, in addition to more powerful computer hardware, is a significant factor in obtaining faster and interactive renderings.

If we are drawing a T-mesh strip, culled triangles will break up the T-mesh run and slow down the rendering on a high performance graphics workstation. When the number of triangles to be drawn exceeds 30,000, we start losing real time interactive animation ability. Even at an ultimate triangle draw rate of 1,000,000 triangles per second for a high end Silicon Graphics machine, it is impossible to draw that many triangles in less than 1/30th of a second video refresh time. This makes interactive rendering problematic, and other techniques of reducing triangle bulk are required. Such algorithms include triangle decimation as developed by Schroeder *et al.* [102].

However, for physical property modeling, we can not assume that a planar geometry does not represent important features in the underlying physics. Figure 8.9 illustrates a calculation of the magnetic field produced by a dipole. The magnetic field is measured perpendicular to a plane of triangles. The calculation is made at

the vertices of each co-planar triangle. Clearly, excessive decimation would obscure important features in the magnetic contours.

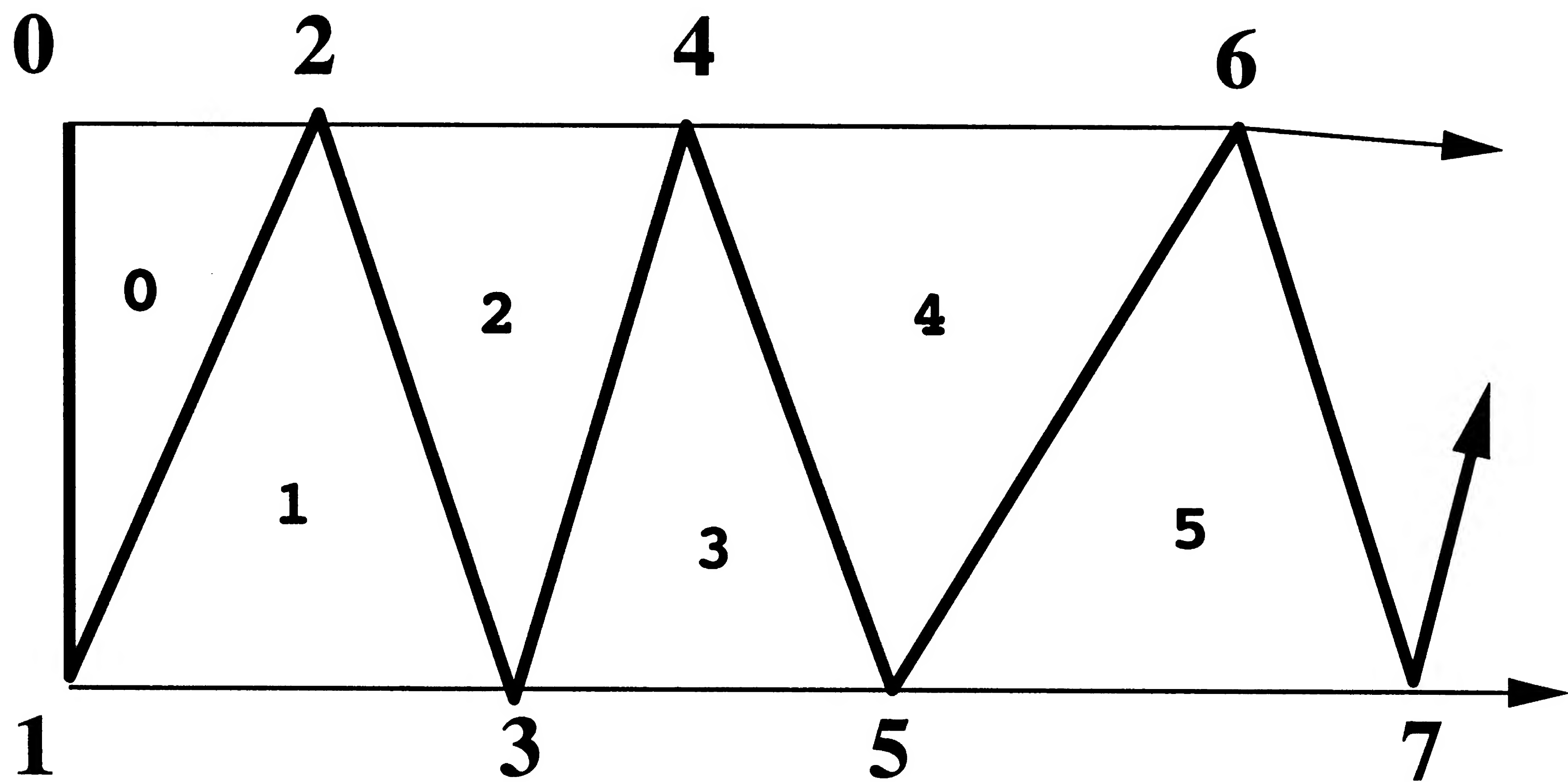


Figure 3.45: Triangles vertices drawn in T-mesh order are drawn fastest.



## 3.2.4 Properties of SpiderWeb Surfaces

### 3.2.4.1 3D Noise and the Surface

Figure 1.9 shows a SpiderWeb A surface generated in random noise volume. This is part of an experiment to see if it is possible to “tear” holes the SpiderWeb surface. The results of the interactive tests show that the SpiderWeb surface does not tear, but it can “writhe” out of the bounding box to produce unresolved edges.

### 3.2.4.2 3D Gradient Strength of Surface

As we have seen in Figure 3.19 the same surface can be generated from a strong or a weak gradient. The SpiderWeb A algorithm generate bubbles, and we assert that bubbles and noise are related. This would imply that a noisy SpiderWeb A surface would have bubbles. Additionally, the bubbles would be associated with regions where the surface was “weak”. We would like to test the hypothesis that noise or a weak gradient is associated with bubbles.

Figure 3.46 shows preliminary results with a noisy three dimensional surface. This Figure is showing experiments to test the relationship between gradient strength and bubbles. The color of the surface is determined by the vertex value difference across the inside to outside voxel edges. A white surface has a strong gradient, and a dark surface has a weak gradient. The thick (green) lines are non manifold bubble triangle edges.

The preliminary results are ambiguous, and it is not clear that the SpiderWeb A bubbles are solely associated with noise. I now believe that the controlling factor for bubbles is the curvature of the local surface and the presence of unresolvable boxel faces that represent a point singularity in the surface as some threshold. This is discussed in Section 3.2.5.

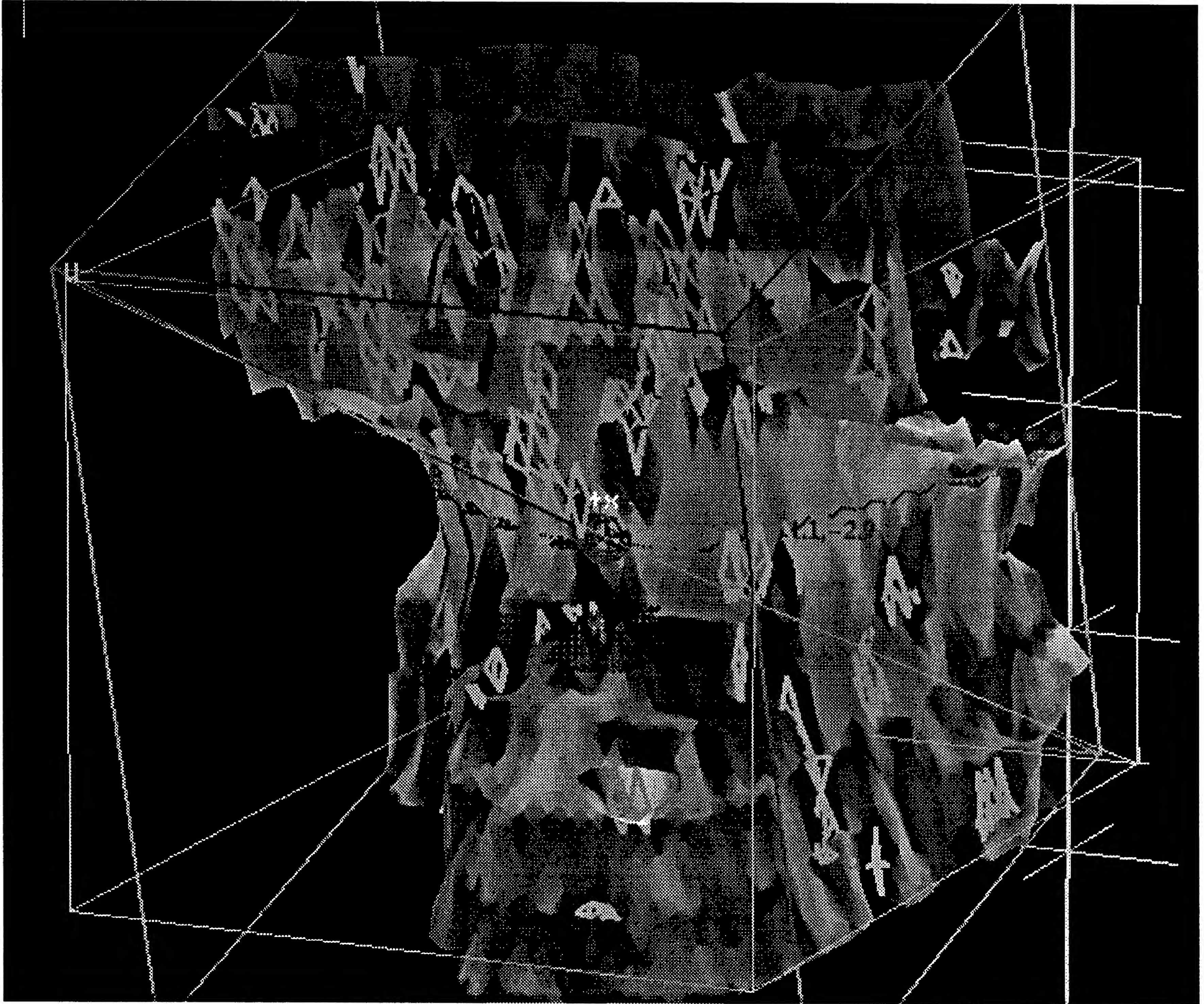


Figure 3.46: A noisy surface just below the skin about the bridge of the nose. The grey color of the surface is proportional to the magnitude of the gradient. The thick (green) lines are bubble bifurcation edges.

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

MISSING

PAGE

MISSING

PAGE



MISSING

PAGE

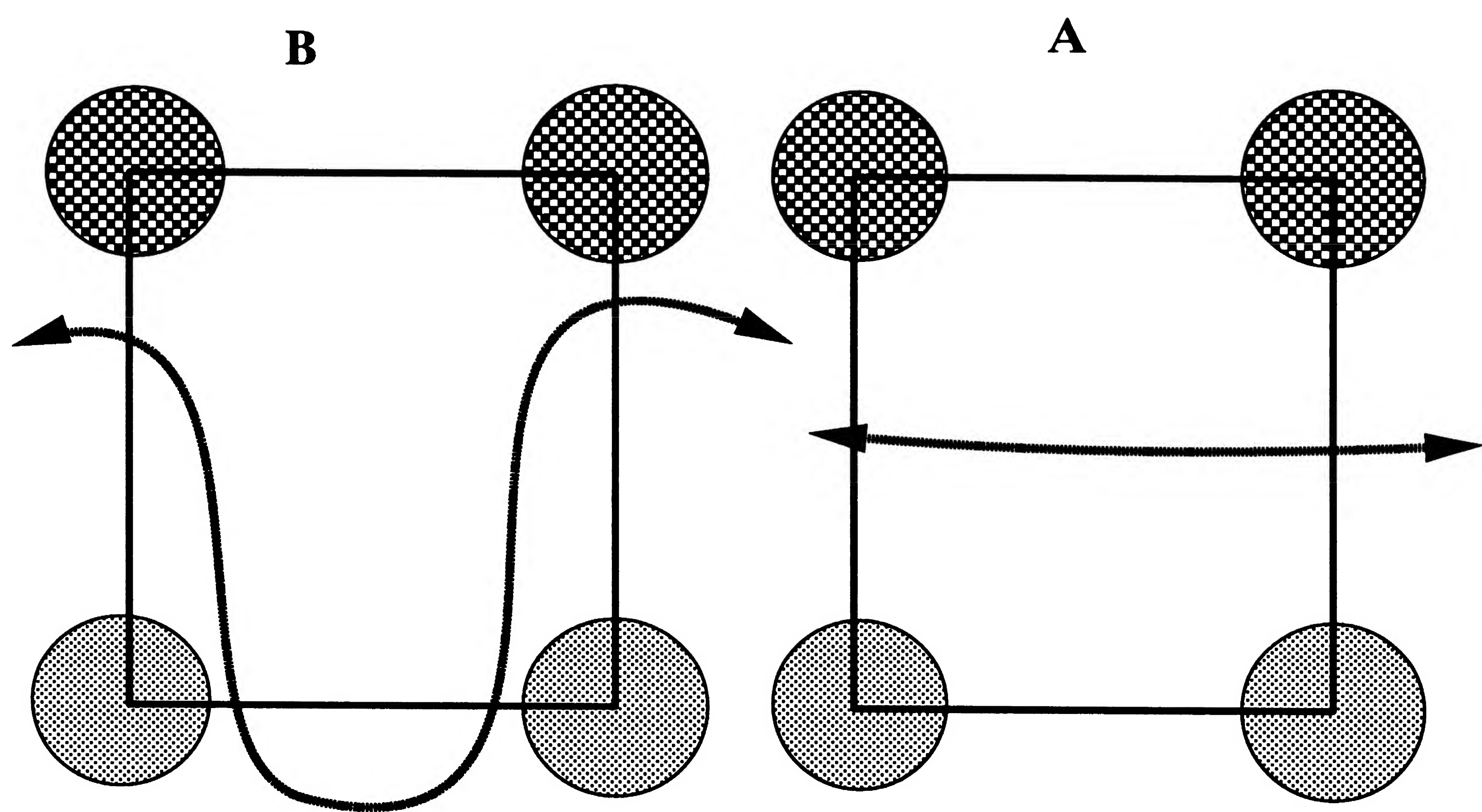


Figure 4.1: Indistinguishable isosurfaces.





## **Part III**

# **The Pointers and Object Registration**



## **Chapter 5**

# **Implementation of the CoordAligner and the Pointers System in Hardware and Software**





## 5.1 Introduction to the “CoordAligner” and the “Pointers”

Precisely locating objects and features in the human body is a fundamental problem in medical science and engineering. Consider the problem of precisely locating a tumor in the brain from computed tomography (CT) slices, and then aiming an intense narrowly focused beam of ionizing radiation, such as from a Gamma Knife, at that tumor. Accuracy and precision are paramount in this situation. The current techniques for precisely aiming the Gamma Knife include bolting a rigid stereotactic halo frame directly to the patient's skull (through the skin) and then the patient's head and frame are scanned. First the tumor is located relative to the halo frame, and then, a treatment plan including radiation beam trajectories are determined. The patient with the halo, is bolted into a Gamma Knife machine. Then, the tumor location, relative to the halo, is irradiated by a gimbaled Gamma Knife source beam rotating about the tumor at the focus center of rotation. There are many other situations where localization of objects in the human body is important.

Measurement of magnetic fields from the brain enable researchers and clinicians to locate neuronal activity related to specific brain processes within specific geometric regions with respect to external landmarks. This procedure is known as Magnetic source imaging (MSI). The process of recording magnetic signals from the head is called magnetoencephalography (MEG). Traditional magnetic resonance imaging (MRI) gives us static anatomy. Magnetoencephalography (MEG) gives us functional information. Our problem is combining the functional and anatomical imaging technologies. A non-invasive method to establish a reliable, precise, and accurate coordinate system is needed for both clinical and research use. This chapter focuses on techniques for combining the two modalities: MRI for anatomic structure and MEG to produce what has been called a functional signal. The techniques that are developed for multi-mode image registration can be directly extended for use with x-ray computed tomography (CT), proton emission tomography (PET) [137], and three dimensional ultrasound [138] [75] [139].

An head-based coordinate system is derived from key defining points, lines, or surfaces on the head. We use the term “Cardinal Landmark” for a point used to define the coordinate system. Precise and accurate determination of the location of these cardinal landmarks is the main purpose of the CoordAlign software system. The “CoordAligner” is a computer program that combines (aligns) MRI stacks of slices and “SpiderWeb” isosurfaces [140] [141] into the Magnetic Source

Imaging (MSI) coordinate system. This permits interactive visualization of magnetic sources within their MRI derived anatomic context.

Two methods were developed to locate cardinal landmarks, one based on the “Pointer” system described here, and the other based on the “SpiderWeb” isosurface system [140] [141]. The SpiderWeb uses a steerable bug to probe the isosurface and find the cardinal landmarks that lie on the surface. The transformation from one coordinate system to another based on the cardinal landmarks is done by the alias transformation matrix in the “CoordAligner”. Various experiments were conducted to establish nominal limits for accuracy and precision of the cardinal landmarks and the registered volume data. The nature and properties of various classes of errors are discussed. Issues and application of image registration in widely different industries and settings are examined.

The SpiderWeb algorithm is a new technique to generate isosurfaces from volume data. It is noise resistant, and produces topologically “correct” surfaces. These surfaces can represent various organs in the body. In particular, we use the skin surface, as reconstructed by the SpiderWeb algorithm, to locate cardinal landmarks in MRI data stacks.

## 5.2 Background and State of the Art of Magnetic Source Imaging

The first detection of the brain’s magnetic field was reported by Cohen in 1968 [142]. He monitored the voltage induced in a multi-turn coil of copper wire and used signal averaging techniques triggered by the electrical alpha rhythm to bring the signal above noise background level. It was not until the introduction of the much more sensitive Superconducting Quantum Interference Device (SQUID) to biomagnetic studies [143] that the first map of a alpha rhythm field was compiled. The magnetic counterpart of the event related potential (ERP) was reported in 1975 by Brenner *et al.* [144] and Teyler *et al.* [145] for visual stimuli.

Brenner *et al.* [146] were the first to demonstrate that it is possible to determine the location of an active region of the brain by analyzing the neuromagnetic field associated with that activity. The activity was evoked by electrical stimulation of a finger. The field pattern exhibits two domains of the field: one where the field emerges from the head (as a positive field), and one where it returns (as a negative



field). Since field lines are continuous loops, without beginning or end, the field must curve around inside the head to close the loop. The positions of strongest field in the positive and negative domains are called field “extremum”. The strength of the nominal magnetic field normally oriented to the scalp at an extremum is on the order of 100 fT ( $100 \times 10^{-15}$  tesla). Depending on conditions signals more than 10 times stronger or weaker have been observed. In contrast the earth’s magnetic field is typically 70 microtesla ( $70 \times 10^{-6}$  tesla), roughly a billion times stronger.

Biomagnetic fields have been observed from many organs of the body since the studies of Baule and McFee in 1963 [147]. They first revealed the magnetic field of the heart. The neuromagnetic fields of the brain must be counted among the weakest of these phenomena, principally because of the weak electrical currents that produce them. Cardiomagnetic fields are much stronger, but because the heart is in constant movement, and the source currents in the heart muscle move about in the chest, locating these stronger signals is more complicated. The heart position must be estimated from 3D time phase models to relate the phase of the heartbeat with the position of the heart.

The accuracy of locating a confined neuronal source has been evaluated through studies of two types of carefully designed phantoms consisting of a Lucite sphere and a plastic human skull cast [148]. The phantoms contained a conducting saline solution in which an electrical current dipole was placed at a known position. Measurements were made with field strengths at slightly above physiological intensities of 500 fT. The deduced source positions were compared with X-ray data to determine the accuracy of the magnetic studies. Fifteen separate procedures with the sphere and ten with the skull yielded source positions that all were closer than 3 mm to the actual source. This demonstrated that MSI has the capability for locating neuronal sources with comparable accuracy.

Magnetic studies of auditory cortex show that observed field strength and location of activated units of a steady-state response vary with tone frequency, while neuronal response magnitude does not [149]. This work suggests that the neuronal population responding to a particular tone is essentially the same size, within a frequency range of 100 to 5,000 Hz. These findings demonstrated that non-invasive methods can establish the functional organization of neuronal populations with respect to brain anatomy. Recent data indicate that the magnitude of the trans-cortical current is 50 picoamperes per square millimeter of cortical area [85].

Electrical activity of the brain produces a magnetic field in the space surrounding the head as well as a distribution of electrical potential across the scalp.



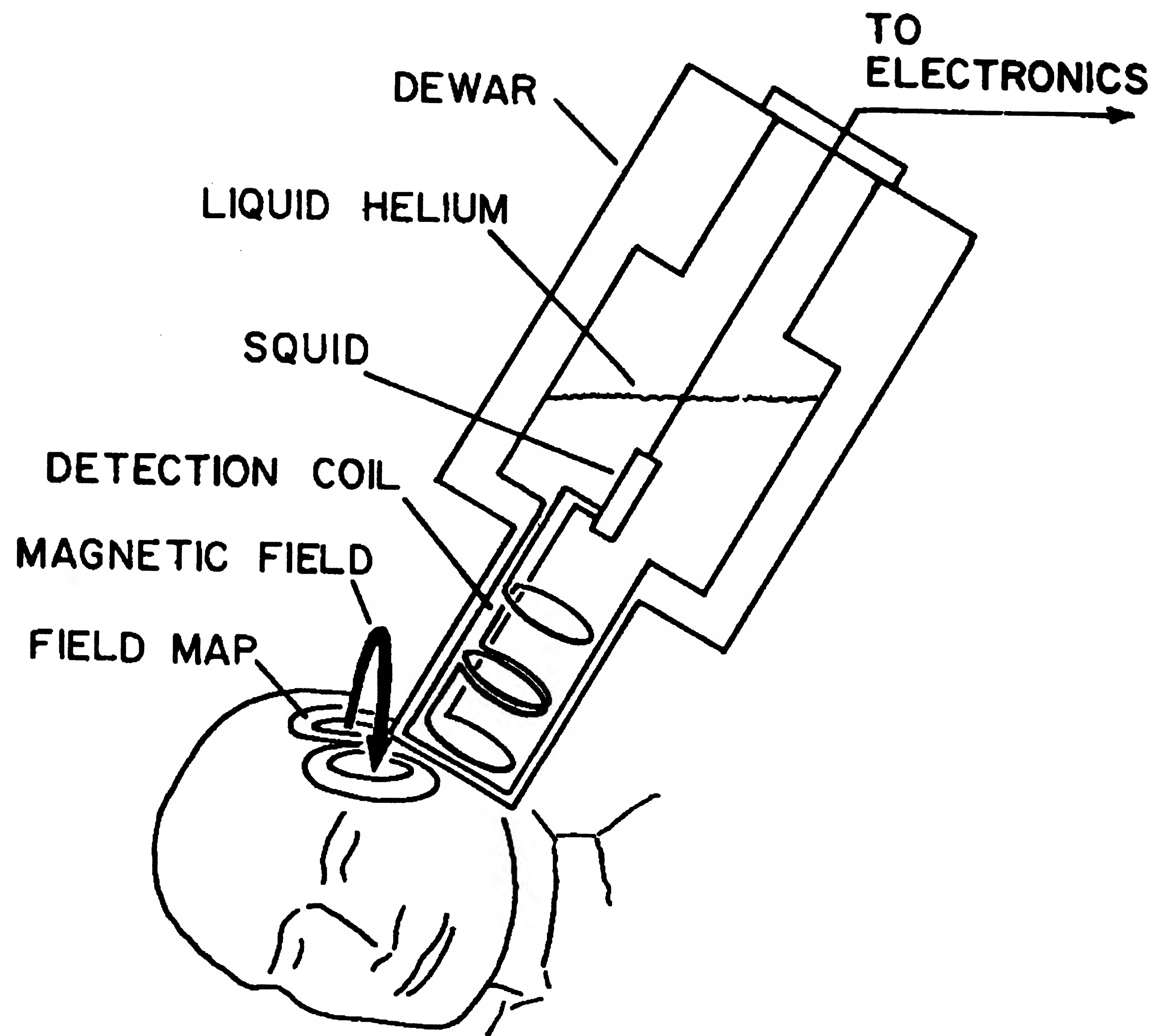


Figure 5.1: Neuromagnetometer measurement of the brain's magnetic field.

Typical magnetic field patterns from a cortical current dipole shows the inflection region of the field lies on or a fissure. The source of such a field pattern is well represented by a short segment of current, or current dipole, positioned under the center of the pattern and directed at right angles to the line between the positions of most intense field extrema in the two domains. This current source thus lies perpendicular to the fissure, as expected if the current is directed perpendicular to the cortical surface within the fissure. The depth of this source can be deduced from the distance across the scalp joining the center of the two field extrema[150][151]. The determination of the depth of the source is independent of the conductivities of the intervening tissue, so long as the classic concentric spherical shell model for the head is valid. Figure 8.7 and Figure 8.8 in Section 8.2 illustrates sample data of the magnetic field pattern over the auditory cortex using computer graphics techniques. For most measurements, of axis of the SQUID dewar is kept perpendicular to the scalp so that the axis of each detection coil is normal to the scalp.

Advances in the techniques for measuring and analyzing the magnetic field produced by the brain may well enhance our ability to locate populations of active neurons whose locations have not yet been determined from scalp potentials alone. Within the context of the concentric spherical shell model for the head, it is possible to determine not only the location of a signal source, but also its magnitude. This can be done because the magnetic field outside of the scalp is independent of the radial dependence of the conductivity, unlike the electric potential at the scalp. This ability to estimate source strength provides a meaningful gauge of the underlying physiological activity.

### **5.2.1 Brief Overview of SQUID Technology**

An ultra-sensitive detector of magnetic fields has been developed based on a superconducting Josephson junction. It is called the Superconducting Quantum Interference Device, or SQUID. The SQUID is maintained at a temperature of 4.2 kelvin by liquid helium in a high vacuum insulated dewar, or by a helium Joule-Thompson refrigerator.

The key element of the system is the detection coil, consisting of superconducting wire loops. Detection coils come in many configurations to suit specific purposes. One common configuration in clinical settings is called a "second-order gradiometer". It consists of three coaxial individual coils wound in series, the center of one being wound in the opposite direction with twice as many turns as each of the

end coils. A magnetic field applied to the detection coil induces a current of superconducting electrons which flows along its leads to a second superconducting coil in series with the first. A property of such a closed superconducting circuit is that the current changes in step with the field so that the net magnetic flux within the circuit does not change. The magnetic flux in a coil is the product of the normal area of the coil and the field  $B$  that is proportional to the area. This current, which is proportional to the applied field, on flowing through the second coil imposes a magnetic field on the SQUID, and the latter's response is sensed by low noise preamplifiers in the head end of the electronics package. Using a feedback technique the output voltage of the SQUID electronics is strictly proportional to the field originally applied to the detection coil. The sensor's frequency response is linear from DC to over 50 kilohertz.

The principal challenge in field measurements is that presented by environmental magnetic noise. The AC magnetic fields from motors, elevators, and subway are many orders of magnitude greater than neuromagnetic fields, and consequently much work has been done to cancel and shield noise. One method is to carry out measurements at a remote site, far from civilization. This has proven successful from a technical standpoint, but it is so inconvenient in clinical practice that such arrangements have been abandoned. The construction of elaborate magnetically shielded rooms enabled Cohen in 1967 to make many pioneering discoveries of a wide variety of biomagnetic effects, but such chambers are expensive, large, and yield a small working volume. Subsequently, the use of detection coils with special noise canceling geometries has proven effective in neuromagnetic studies. The second order gradiometer geometry makes the detection coil less sensitive to the magnetic field from distant sources in which the field is comparatively uniform, whereas it retains sensitivity to a nearby source, such as the brain, whose field is much stronger at the closest coils. The second-order gradiometer permitted many groups to begin MEG research, since a variety of effects can be observed in a normal laboratory or clinical environment without cumbersome shielding.

Over the past decade the sensitivity of magnetic sensors has improved by an order of magnitude, and noise rejection has increased by two orders. A SQUID sensor with a detection coil of 2 cm diameters has a system noise level that is equivalent to a root-mean-square field of less than 10 fT within a bandwidth of 1 Hz.

Determining the magnetic field map across the scalp had been a laborious procedure. It involves sequential measurements at a large number of positions, typically thirty or so for a reasonable resolution. New systems can provide simultaneous coverage of the whole scalp. For instance a 122 sensor system has been developed in Helsinki by Ahonen *et al.* [152] [153]. It is possible to register animated maps of the



magnetic field of the brain in real time[154]. Real-time monitoring of brain function is possible with strong spontaneous activity such as alpha rhythms. Weaker activity, such as brain responses to sensory stimulation need to be averaged together to raise the signal above the background noise

## 5.2.2 Locating a Current Dipole

To explain the relationship between sources and field we use the classic current-dipole model for the most elemental source. This simplified model is adequate when considering a small region of active tissue, where the largest linear dimension of the active region is much smaller than the distance from the region to the field sensor. In the general case of a wide distribution of cortical activity, the superposition principle permits us to view the resulting magnetic field and electric potential patterns as arising from a distribution of current dipoles within the active area, and the field and potential are the sums of the contributions from these individual dipoles. Thus the principles to be established for a single current dipole have a straightforward generalization to situations with multiple sources. The difficulty is in the reverse or dual problem, where we have to estimate the current dipoles from knowledge of the magnetic field.

### 5.2.2.1 Volume Electrical and Magnetic Fields: Law of Biot and Savart

The source of a magnetic field can be identified by the law of Biot and Savart [155] which specifies the contribution made by the current density at each point in space to the field at a given point of observation. Biot-Savart law states that the field  $\vec{B}$  at a distance  $r$  from a current  $\vec{I}$ , flowing along a short path length  $\vec{L}$ , is tangential to a circle centered on a straight-line extension of the current's direction. Consequently, the field lies in a plane that is perpendicular to the dipole. The strength of the field  $\vec{B}$  (the magnetic flux density) is proportional to the number of ions comprising the current and their average velocity. This is equivalent to saying that the strength  $|\vec{Q}|$  of the current source is the product of the current and the length of its path:

$$(5.1) \quad \vec{Q} = \vec{I} L$$

This is called the “current dipole moment” of the source. The strength of the field in any direction varies as the inverse square of the distance  $r$  from the current, as given



by

$$(5.2) \quad \vec{B} = \frac{\vec{Q} \sin \psi}{\mu_0 r^2}$$

This formula gives the field  $\vec{B}$  in the SI units of tesla when  $\vec{Q}$  is expressed in ampere-meters and  $r$  in meters. The symbol,  $\mu_0$  is the permeability of free space and has the value  $\mu_0 = 4\pi \times 10^{-7} \text{ N/A}^2$ . At a given distance, the field is strongest to the side of the current ( $\psi = 90$  degrees), and it diminishes to zero in the direction of the current ( $\psi = 0$  degrees) [156] [157].

### 5.2.2.2 Cellular Currents

Consider the cellular basis for a current dipole model and what aspects of an actual current pattern give rise to a magnetic field. Consider the dendritic tree of pyramidal cells responding to synaptic activity. Synaptic excitation changes in permeability of the cell membrane which allows an influx of positive ions into the cell, leaving a local depletion of positive charge in the extracellular medium. The mutual repulsion of the positive ions within the cell produces an axial current and an increasing positive charge near the head of the dipole arrow. This repels positive ions in the extracellular medium, resulting in a current directed back toward the negative charge. The extracellular current and oppositely directed intracellular current complete a closed circuit. Within the relatively long time during which the current pattern develops in tens of milliseconds, it is reasonable to assume that the intra- and extra- cellular current pattern is approximately quasistatic. Kirchoff's law balances the total current in the extracellular return flow with the intracellular current.

Any small region of the current pattern produces a magnetic field. Therefore, one might expect that the entire pattern contributes to a measured neuromagnetic field. However, only the intracellular driven current is important. Consider the case of the neuron immersed in an infinite, uniformly conducting medium. Furthermore, consider the current pattern as divided into three regions: the intracellular current, represented by a current dipole, the transmembrane current; and the extracellular current. First we note that the intracellular current contributes to the magnetic field within the conducting medium, just as for a current dipole. However, the transmembrane current, contribution to the field must be much smaller for two reasons: (1) in comparison with the intracellular current the volume containing the transmembrane current is much smaller because of the thinness of the cell membrane; and (2) the radial symmetry of the current pattern causes substantial cancellation of the field from current on one side of the cell with the field from oppositely directed

current on the other side of the cell. This was shown mathematically by Swinney and Wikswo in 1980 [158]. It is ironic that transmembrane current, which gives rise to the cellular excitation the first place, plays no appreciable direct role in producing the detected magnetic field.

In a uniform conducting medium the net contribution of the extracellular current is zero. This can be demonstrated by assuming the intracellular current flow is well represented by a current dipole, in which case the extracellular current pattern is dipolar, the same as arises from a fixed positive charge at the head of a dipole arrow and negative charge of equal magnitude at the tail of the arrow. At each point in space the volume current can be specified by its current density, denoted by  $\vec{J}$ , expressing the current per unit area normal to the direction of flow. The Biot-Savart law can be applied to such a continuous distribution of extracellular current by first imagining the volume divided up into small contiguous regions. The volume current density within each region is uniform to a good approximation; and for computing the field produced far away the current distribution in this region may be represented by a current dipole moment  $\vec{Q} = \vec{J}dv$  lying parallel to the current density. The summed contributions from all the individual regions can be converted into an integral over the entire extracellular volume, with  $\vec{J}dv$  replacing  $\vec{Q}$  in the expression. When the Biot-Savart law is integrated in this way over the dipolar pattern of the volume current, the net result turns out to be zero. The contribution from one region of space is canceled by a contribution from another region by symmetry.

The fact that the measured magnetic field is produced by the intracellular current and not by the extracellular current is important in simplifying our considerations of fields sources. Ampere's law says that the line integral of the magnetic field around a closed path is proportional to the net current passing through the enclosed area. It is more usefully applied in situations of symmetry where the field is known to have a constant value along the path. If the closed path is imagined to be a circle centered on the current dipole, then the field is tangential to the circle, and has the same magnitude everywhere, so the line integral is simply the product of the field magnitude and the circumference of the circle. There are two contributions from current passing through the area of the circle: the current dipole and the returning volume current. It does not follow that the field has contributions from both of these current flows. Ampere's law does not specify which elements of the current enclosed within a path of integration contribute to the field: this law follows from the notion of charge conservation, or current continuity, and the laws of electromagnetism, which for a given current source and distribution of conductivity predict the pattern of volume current. Thus Ampere's law is necessarily consistent with the Biot-Savart law, and can be considered the integral form of the law. Because it is formulated



in integral form without explicit reference to current elements, it does not identify which currents give rise to the net field.

### 5.3 Magnetic Resonance Imaging Physics and Technology

MRI slices and 3D reconstruction of brain anatomy, when registered with MSI brain sources, give us a functional anatomy that neither modality yield alone. MSI depends on MRI to show us the anatomy at a magnetic source.

A fascinating history of medical imaging tomography is provided by Webb [159]. The main idea of Nuclear Magnetic Resonance (NMR) was originally described by Rabi in 1939 *et al.* [160] as the *Larmor* or resonance frequency for particle beams. In 1946 Bloch [161] measured this resonance frequency for liquids, and Purcell [162] did the same for solids. They received the 1952 Nobel Prize for their work. The beginning of the current generation of imaging techniques was started by Lauterbur in 1973 [163], who created the first images of a water sample by using magnetic field gradients to obtain spatial information from the echo signals. Mallard went on to develop many techniques for geometric localization of signals leading to many of the modern phase encoding techniques [164]. Damadian [165] first observed that tumors have much longer longitudinal spin-lattice relaxation time ( $T_1$ ) than normal tissues in general. This was the first demonstration that pathological tissue could be clearly contrasted from normal tissue. He acquired the first human images using a prototype superconducting magnet [166]. The first commercial development of MRI was a prototype produced by Damadian who founded Fonar, Inc. on Long Island, and since then there has been an explosion of subsequent research in MRI. An interesting line of research in MRI is the use of SQUIDS for improved signal detection leading to either lower intensity static fields being used or finer details of images being obtained [164].

The NMR signal is based on a set of tissue parameters derived mainly from hydrogen atom density and two relaxation times called  $T_1$  and  $T_2$ . Because it takes a finite time to develop the signal echo from the excited hydrogen nuclei, MRI signals are modulated by moving fluids or bodies. By using these parameters for hydrogen protons, a great deal of clinical information is available from MRI studies. Hydrogen nuclei produce different signals depending on the surrounding tissue. Images based on either fat or water protons can be obtained using chemical shift phenomena. The

components of the hydrogen echo signal are fundamentally different from the signal obtained in transmission X-ray CT. The information is largely based on the X-ray interaction with tissue electrons and the tissue mass density.

The unpaired nuclear particles cause the magnetic moment on the nucleus. Thus,  $^{12}\text{C}$ , having equal numbers of protons and neutrons, can not be magnetized, while the isotope  $^{13}\text{C}$  is magnetized and could be used in MRI (if there were enough atoms to send back a measurable signal). Each nuclear species, being composed of different numbers of protons and neutrons, will have its own total spin value. Depending on these values, there may be a strong nuclear magnetic moment. Nuclei with zero spin like helium and carbon cannot be imaged. Nuclei with an odd charge but even mass have integral spins, like hydrogen, nitrogen, and phosphorus. Because diamagnetic (with an odd number of protons) have a net magnetic moment, they will align themselves in an applied magnetic field. Within any externally applied magnetic field there is always a very small net excess of dipoles aligned in the lower energy parallel direction compared with the anti-parallel orientation. This excess population is called the net magnetization ( $\mathbf{M}$ ). The main reason for using powerful magnets in MRI is to maximize  $\mathbf{M}$ , which results in a greater signal strength.

The radiofrequency (RF) pulse represents the first step of the process by which  $\mathbf{M}$  generates an echo signal. RF energy photons are absorbed by a proton. This causes it to flip from the lower energy state to the higher energy state with respect to  $\mathbf{M}$ . The frequency associated with the absorption energy is known as the Larmor or resonance frequency. It is a function of field strength and the type of nucleus. The Larmor frequency is given by:

$$(5.3) \quad f_{\text{Larmor}} = \gamma B_0$$

where  $f_{\text{Larmor}}$  is the resonance frequency in MHz,  $B_0$  is the applied magnetic field strength in tesla, and  $\gamma$  is the gyromagnetic ratio, with the units of megahertz per tesla. The group magnetized population,  $\mathbf{M}$  is randomly oriented. The presence of the strong static field  $B_0$  polarizes the group  $\mathbf{M}$  of the dipole population parallel with  $B_0$ . By convention, the z-axis is defined by the direction of the static field  $B_0$ . The plane perpendicular to the main field is the transverse, or x-y plane. Prior to the application of an RF pulse, the protons are aligned along the  $B_0$  or z direction. In the classic view, the RF pulse is viewed as an oscillating magnetic field. This magnetic field, called  $B_1$ , is produced by an RF transmitting coil. The transmitter coil is oriented so that its  $B_1$  is in the transverse plane, perpendicular to  $B_0$ . When the RF pulse is applied, the effect of  $B_1$  is to rotate  $\mathbf{M}$  away from its equilibrium alignment along the  $B_0$  or z-axis, while remaining in the plane perpendicular to  $B_1$ .



The spherical angle by which the RF pulse rotates  $\mathbf{M}$  away from the  $z$ -axis is called the flip angle. The flip angle increases with the power of the RF pulse. Any flip angle can be produced. The choice of flip angle depends on the particular imaging technique desired. For instance, in spin-echo pulse sequences, a 90-degree RF pulse rotates  $\mathbf{M}$  completely into the transverse plane. Small flip angles are used for gradient-echo pulse sequences, so only a small component of  $\mathbf{M}$  lies in the transverse plane. This transverse component of  $\mathbf{M}$  is responsible for the production of the MR signal.

The flip angle is a polar angle, and  $\psi$  in the transverse plane describes the “wobble” or precession away from the  $\mathbf{B}_0$  axis. This wobble of spinning charged protons produces a RF echo signal, the protons shed the energy they picked up from the RF pulse and the wobble collapses. As  $\mathbf{M}$  precesses around  $\mathbf{B}_0$ , it induces an RF signal in an antenna coil whose axis lies in the transverse plane. The precession rate determines the frequency of the induced voltage. This collapse or re-alignment of protons with the  $\mathbf{B}_0$ , after being perturbed by the RF pulse  $\mathbf{B}_1$  field produces the rephasing echo signal.

The spatial distribution of rephasing protons can be measured anywhere within the magnet bore by manipulation of the frequency and phase of an additional gradient magnetic field. The gradient magnetic field  $G$  is a weak magnetic field but aligned parallel with the main field  $\mathbf{B}_0$ , at most a few percent as strong as the main  $\mathbf{B}_0$  field. It is produced by DC current pulses in coils positioned within the main bore so that the overall magnetic field is not uniform within the bore. The gradient  $G$  produces a position dependence ( $\delta F$ ) in the Larmor resonance frequency expressed as:

$$(5.4) \quad \delta F = \gamma(G \cdot x)$$

where  $x$  is the position along the gradient. When the gradient field is superimposed on the main field, the Larmor frequencies assume values that depend on the position in the gradient, as expressed by

$$(5.5) \quad f_{Larmor} = \gamma \mathbf{B}_0 + \gamma(G \cdot x)$$

The mapping of the Larmor frequency shift into source signal shift is provided by various inverse Fourier transform techniques in two and three dimension. The design of MRI imaging techniques depends on designing RF pulse sequences, and gradient electromagnetic sequences to balance sensitivity and high spatial resolution.

The position of an echo in three dimensions is encoded by the frequency and phase information in the echo signal.

- The first gradient magnetic field encodes the slice position. This is the slice selection gradient.
- A second gradient encodes one dimension of the echo location in the plane.
- A third gradient encodes location in the plane dimensions in the phase of the MR signal. This is also known as twisting of the phase, or “spin warp” encoding.

For convenience, the slice selection direction is referred to as  $z$ , the frequency encoding direction as  $x$ , and the phase encoding direction as  $y$ . However, this designation is arbitrary and is unrelated to the physical orientation of the gradient coils. The  $x, y, z$  coordinate system is oriented by the operator, and is entirely arbitrary. The relation between the MRI coordinate systems and other medical imaging modalities such as MSI is the main research focus of this chapter.

Slice thickness is controlled by the pulse bandwidth spread of slice selection frequency. This spread depends on the strength of the slice selection gradient and the bandwidth of the RF pulse. The slice factor is the fraction of the slice interval covered by the slice thickness. As the slice selection gradient strength is increased, the range of frequencies across the slices also increases. As a result an RF pulse with a fixed bandwidth will produce fewer spins at a fixed frequency, and the effective signal from the slice volume is decreased.

### 5.3.0.3 MRI Pulse Sequence Overview

Different patterns of RF and gradient coil pulses can be used to produce different types of MRI echo signals. Typically, these pulse patterns are computer programmable. Certain broad classes of pulse programs are used to image different anatomic features. We will briefly describe at some of the more popular pulse sequences.

Free induction decay (FID) is the signal that is produced immediately after an RF pulse. This signal is not used in MR imaging because time must be allowed for the gradient magnetic fields to be applied to localize the echo. To generate an MR signal, the echo at certain times after the FID is used.

Spin Echo consists of two RF pulses, a 90-degree and 180-degree pulse separated by equal intervals of the transverse echo  $TE/2$ . The 90-degree RF pulse excites the protons and generates an FID signal. The 180-degree RF pulse refocuses



the precession in the  $x$ - $y$  plane  $\mathbf{M}$  so that the dephasing effects from the static magnetic field inhomogeneities, due to magnet defects, bad shimming, or local tissue differences in magnetic susceptibility, are canceled. As a result, the image is an average that is weighted more for the second pulse ( $T_2$  weighted).

Gradient echo pulse sequences use only a single RF pulse that results in a less than 90 degree flip angle. The formation of an echo with these sequences requires two gradient magnetic field pulses. The application of the frequency-encoding gradient field during the echo immediately dephases the spins along the local  $x$ -axis. To correct this dephasing, an inverted gradient pulse is applied, which produces a compensating phase shift in the opposite direction. The first gradient pulse is the dephasing gradient, the second pulse is the rephasing gradient. The pair of dephasing and rephasing pulses constitutes a gradient reversal, which results in the formation of a gradient echo.

Inversion recovery has three RF pulses and consists of an 180-degree RF pulse followed by a spin echo pulse sequence. The initial 180 degree pulse inverts  $\mathbf{M}$ . Then there is a TI inversion period in which the protons re-magnetize. After this inversion period, an MR signal is produced by using a conventional spin-echo pulse sequence.

There are a number of other classes of RF and gradient field pulse sequences commonly used in clinical scanning. Each technique generate different contrasts between different types of tissue. A review of clinical pulse sequences is given by Edelman and Hesselink [167].

### 5.3.1 The Problem of Disparate Coordinate Systems in Medical Imaging

The main problem addressed in this chapter is to express both the functional image obtained with SQUID measurements and anatomical image from MRI in the same coordinate system. We have seen in Section 5.3 that the MRI orientations are arbitrary, and need to be transformed into a common coordinate system with the locations from the SQUID measurements. Mathematically, this is accomplished by an “alias” transform of the coordinate system of one imaging modality into a common system. This permits the same  $x, y, z$  coordinates to specify the same location in both an MRI stack and MSI. An alias transform is a matrix that converts one coordinate system into another. It is a transformation that rotates, translates, and

possibly scales the coordinate system. It operates only on the coordinate system, not the locations specified by these coordinates.

A stack of MRI slices has a coordinate system that is defined in terms of the MRI magnet bore, not the head of the subject being examined [167]. Different MRI manufacturers use different conventions for the chirality of the coordinate system. Some use a right handed coordinate system, and others define a left handed coordinate system. The observer's point of view can be considered to orient the handedness of the coordinate system. For example, a coordinate system used in radiology is defined from the hands (the point of view) of the radiologist, and not of the subject. The Z axis is called the Cranio-Caudal (CC) axis, and increases in value toward the cranial (top of the head, or apex) direction. The X axis is the Anterior-Posterior (AP) axis, with X values increasing in value toward the anterior direction (in the direction of the chest and face). The Y axis is the Left-Right (LR) axis, and it increases in value toward the right hand. This creates a *Left Hand* coordinate System (LHS). This convention is used in the Philips GYROSCAN<sup>TM</sup> 1.5 T MRI machine at New York University Medical Center, which has a left hand coordinate system [168]. Neuromagnetometer coordinates are in the head-based coordinate system, defined by the three cardinal landmarks of the head, as described below. The +Z and +X axis are again oriented along the cephalic and anterior directions, respectively. However, +Y is oriented toward the subject's left hand, creating a *Right Hand* coordinate system. This can be seen by playing with the three fingers of the right or left hand [169] [170] [171]. The thumb points along +X, the first finger points along +Y, and the third finger points along +Z. For a Right Hand coordinate system, the first finger points Left, and for a Left Hand coordinate system, the first finger points right. We use a Right Hand coordinate system where the right hand is along -Y direction. Errors in chirality (handedness) in medical imaging are common. Many benchmark medical image data sets are rendered reversed, mirrored, or backward (the Y values reversed) by different rendering algorithms.

#### 5.3.1.1 Coordinate System Aliasing

Cardinal landmarks are precisely defined and easily identified locations on the skin of the head that are used to define the head-based coordinate system used for MSI. A head-based coordinate system is defined by three or more non-colinear landmarks. The precision and accuracy of the registration of different coordinate systems is only as good as the location values obtained for the cardinal landmarks in each of the different coordinate systems. The process of locating a point on a stack of image



slices is called “sighting”. This is directly analogous to a navigator taking sighting on landmarks or stars from sea to get compass bearings. However, sighting biological landmarks is not as well defined as sighting stars or mountain peaks. We have developed various definitions to assist in reliable and precise reading of cardinal landmarks. The precision in sighting is the variation in repeated measurements of the same location. High precision measurements have a relatively small standard deviation in many measurements. Accurate sighting is the systematic, repeated error in locating a cardinal landmark with respect to its actual position. A calibration error is such a systematic error in accuracy. Precise and accurate localization of the cardinal landmarks depends on how we choose and define our landmarks, and the topic we shall next address.

A fiducial mark is a location that is used to align objects with respect to each other. Its principal properties are that a fiducial can be located precisely and accurately. However, they may not have any intrinsic importance on the object being registered. Locations of high surface curvature are easy to repeatedly and precisely locate on the head, and have value as fiducial marks. When a small number of fiducial marks are used to define a coordinate system, we call them cardinal landmarks. A cardinal landmark is typically on a line or point of curvature maximum [172] on the skin surface that is easy to accurately identify and can be precisely re-located by pointing at it. David Hilbert [135] in 1932 suggested that curvature maxima were important facial features. He illustrated his idea with marble sculpture busts. This idyl observation was noted by Koenderink [120], who bases Hilbert’s description of facial features by curvature on a conjecture proposed by Felix Klein. Neither Hilbert or Klein did not follow this idea further. Thirion *et al.* [173] [129] [174] have developed a method to delineate curvature maximum points and lines on reconstructed skull surfaces. Many landmarks are associated with the crest lines, but the relationship between cardinal landmark and crest lines of curvature maxima is not straightforward.

Kaufman and Williamson [175] [151] conceived a modified cartesian coordinate system based on the plane defined by the nasion, and the left and right periauricular points. This coordinate system is calculated from the location of three cardinal landmarks, and they define the “PPN head-based coordinate system”. In short hand this is known as the “PPN Headframe”. This coordinate system has a natural relationship to the conventional ten–twenty system for defining electrode positions on the scalp [176], except that the ten–twenty system employs the preauricular instead of the periauricular points. The former cannot be located as accurately as the latter, which is why MSI is based on the latter points. The complete algorithm is given in Section 5.4.2, and sample code is given in the Appendix, Section A.1. These cardinal landmarks are precisely located within a 1.2 mm standard deviation by an

experienced operator, as shown in Table 6.2. Biomagnetic Technology, Inc. (BTI) has adopted this system for aligning the subject's head to the neuromagnetometers sensors [177]. The cardinal landmarks are sighted on the head by an operator lightly touching a Polhemus [8] Navigation, Inc., 3Space digitizer probe to the skin features [177] [178]. These landmarks are used as the reference frame to register the position of the Neuromagnetometer SQUID sensor coils to the head. From the known position of the SQUIDs, we can determine the position of magnetic field recording, and the position of the brain magnetic sources with respect to the head. The PPN head-based coordinate system is uniquely defined for each subject by the location of the cardinal landmarks on the subject's head.

#### **5.3.1.2 Locating landmarks in MRI slices**

We have seen that a head-based coordinate system can be constructed from the cardinal landmarks. In this section, we examine issues in exactly how we obtain  $x, y, z$  values for these cardinal landmarks, and techniques to increase the precision and accuracy of these critical measurements. A software program, called the CoordAligner, was written to interpret MRI stacks of slices and render precise measurements from them. The software support three techniques we use to precisely identify cardinal landmarks. The techniques are the direct mode, the Pointers technique, and the SpiderWeb technique.

#### **5.3.1.3 Direct Estimation of Cardinal Landmarks on Slices**

The simplest and most commonly used method of identifying a cardinal landmark from MRI slices is by using a Vitamin E capsule taped to the subjects head over each landmark. The blobby trace on a slice from each pill is used to approximate where the capsule touches the skin and thereby the location of the landmark. Clearly, this method provides at best an estimation of the cardinal landmark location. This direct method is used as a baseline to gauge the precision and accuracy relative of the other two methods described below.

A software system called the CoordAlign software was developed to locate landmarks from MRI stacks and slices. It also establishes the PPN Headframe for importing MSI data. The user can select any slice in any stack for viewing in the slice viewing window, dubbed the "PixelPicker". When in the "direct locator mode",



a mouse click on a slice image is transformed from the two dimensional slice location into a 3 dimensional location on the slice in MRI space or the PPN Headframe. This location is used to position a marker of the cardinal landmark. However, the marker can not be interpolated above or below the plane of the slice. This limits the ultimate accuracy of the direct locator mode to that of the inter-slice spacing.

#### **5.3.1.4 The Pointers for Cardinal Landmark Localization.**

Generally, a cardinal landmark will not lie precisely in the plane of any one MRI slice. Precise, three dimensional localization of a landmark must include a facility to interpolate its position between slices. Ideally, we would like to be able to locate a landmark independent of the slice position or orientation relative to the true location of the landmark.

A solution to the problem of interpolating landmark locations between slices is the “Pointers” technique. The Pointers are a set of external markers that provide a set of “landing approach lights” to permit the user to home in on a cardinal location. The Pointers make a converging pattern of marks in a MRI stack of slices that converge the apex of the pointer tip, where it rests on the skin of the head.

Each pointer is small solid Plexiglas<sup>TM</sup> cone through which 4 hollow cylindrical spokes of 5 mm diameter are blind drilled. The conical body of the Pointers is approximately 4 cm in size. The spokes are filled with Vitamin E, a long chain fatty material that is traditionally used as a MRI marker because it produces a bright MRI signal. The center lines of the Vitamin E spokes converge slightly beyond the apex of the Pointer cone, and provide a means to interpolate the apex position from the slices that intersect the body of the Pointers. Figure 5.2 and 5.3 shows a computer model of the Pointers. Figure 5.4 shows how they are positioned over the right periauricular point. The Pointers are attached to a Velcro headband, or, alternatively, to a rigid plastic welders headband. Figure 5.14, 5.15 and 5.16 shows the pointer traces as they appear in the CoordAligner software system. The Pointers provide multiple traces in the stack that allow precise interpolation of the intended landmark location between slices. Figures 5.17 and 5.18 illustrates the system of radial spoke traces in a stack of MRI slices is interpolated into a cardinal landmark. The Pointers are precisely positioned over the cardinal landmarks on a subject just prior to a MRI study.



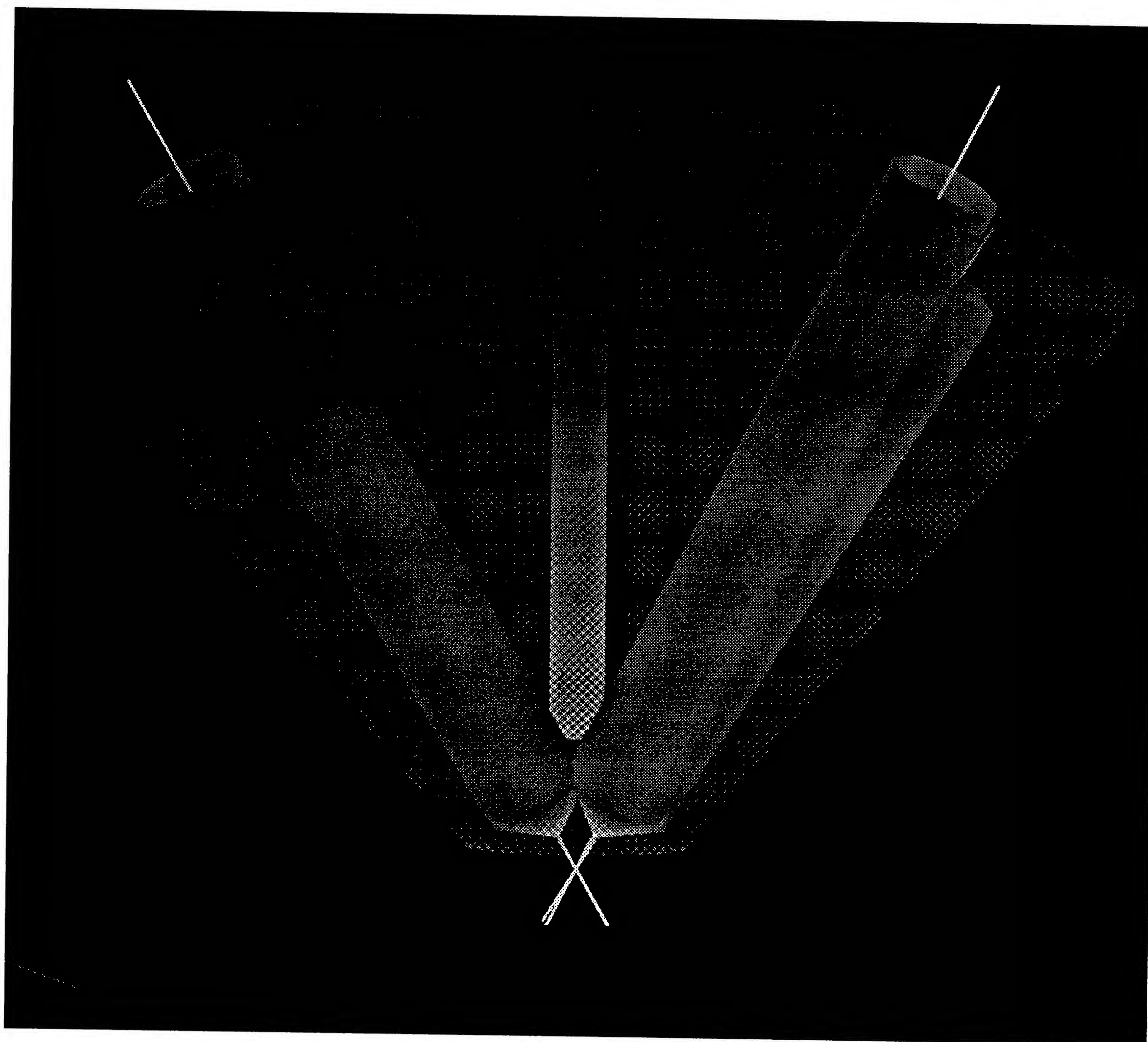


Figure 5.2: Computer model of a Pointer, viewed from the side.

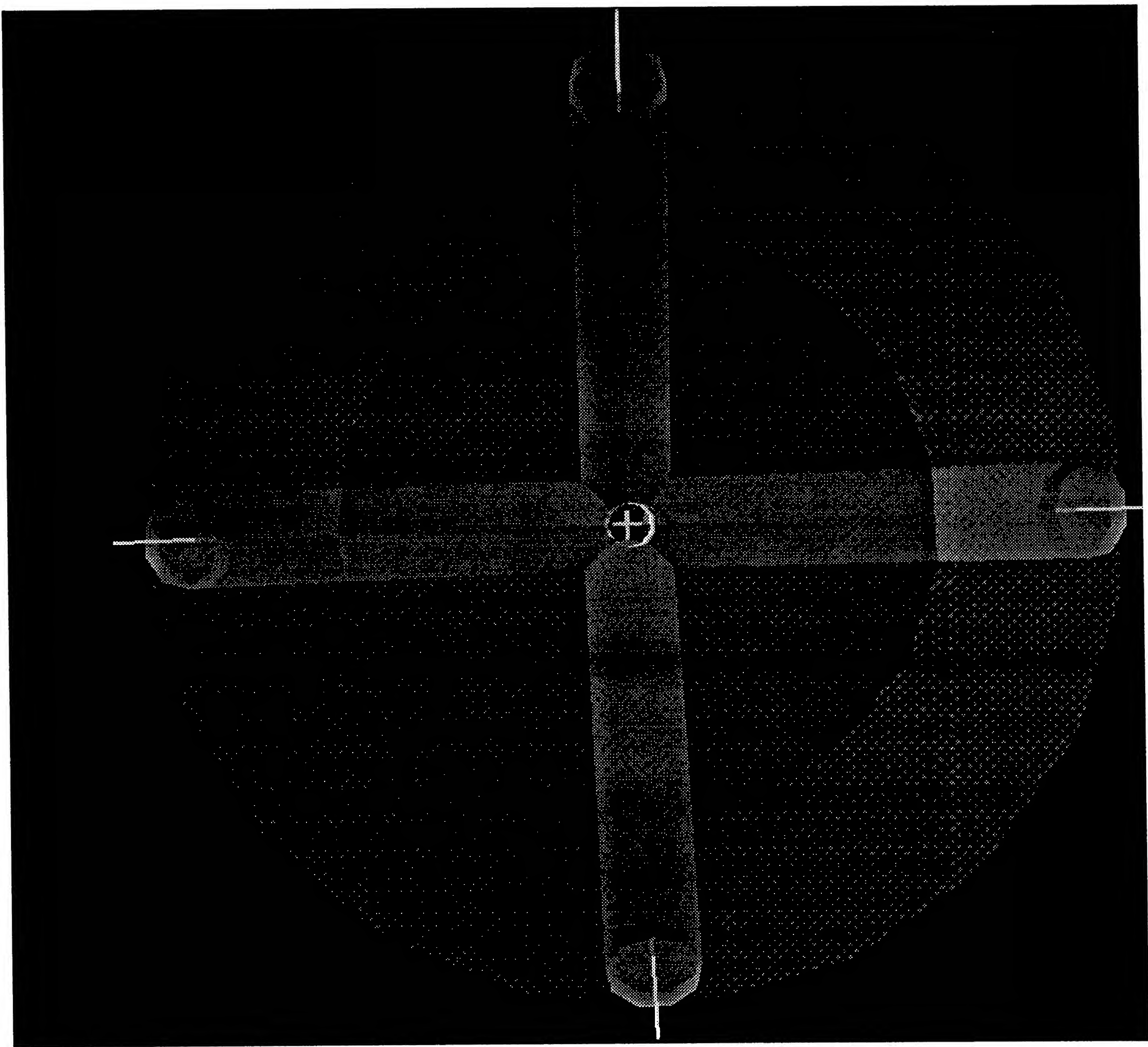


Figure 5.3: Computer model of the Pointers, viewed from the top.



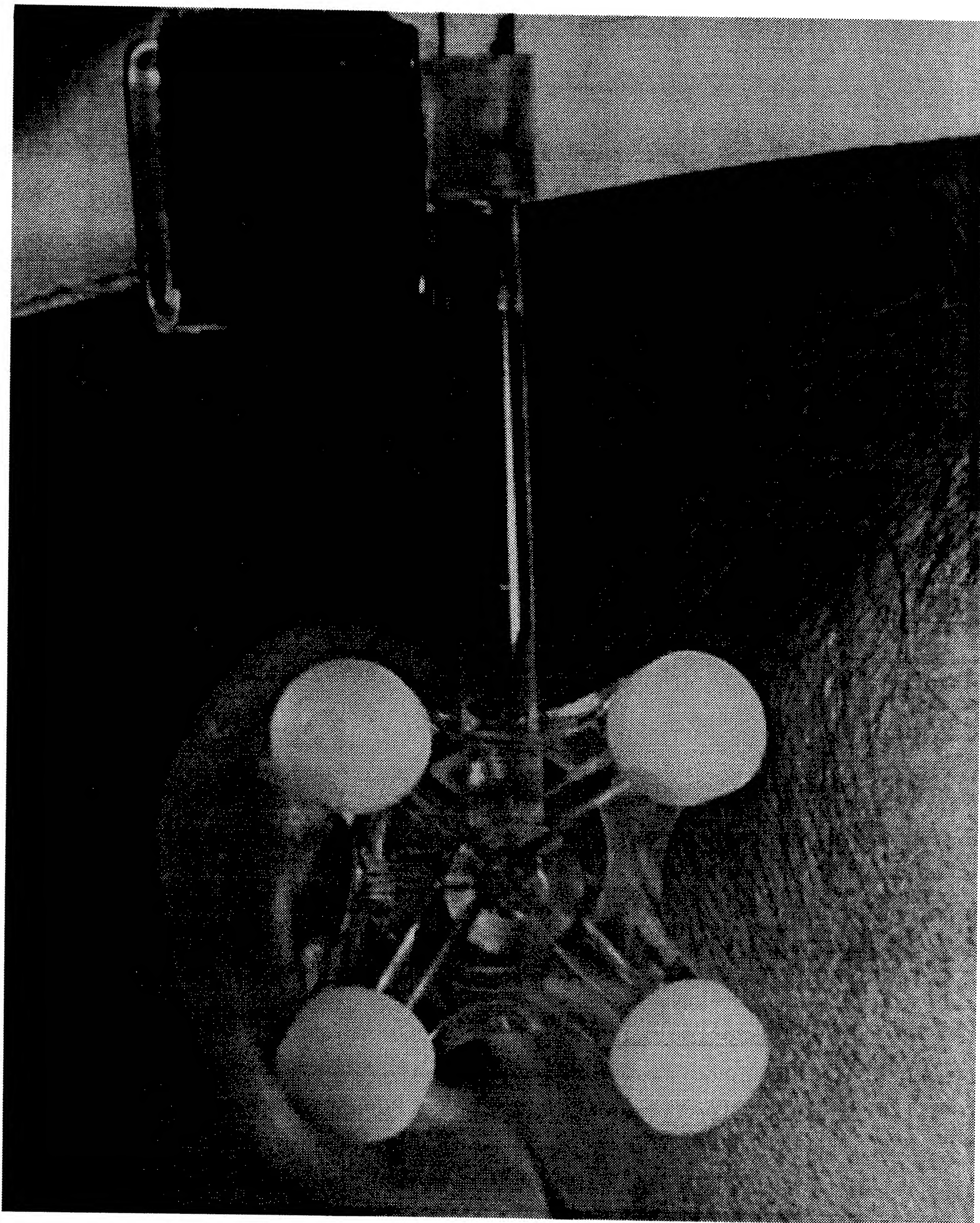


Figure 5.4: A Pointer worn over the right periauricular point.



#### 5.3.1.5 The SpiderWeb for Landmark Localization

Our experience with the Pointers revealed certain drawbacks, as well as advantages. The main drawback of the Pointers is that excessive clearance ( $\simeq 4$  cm) about the head is required once they are positioned on the subject. Then the Pointers must not be bumped or otherwise moved once they are attached and positioned. The entry of a subject into a MRI bore must be executed with great care so as not to dislodge the Pointers. In addition, the new generation of MRI machines have close fitting, helmet like head coils that make use of the Pointers problematic. One possible solution is to build the Pointers into a head coil helmet.

MRI slices contain information that can define the skin air interface. This skin surface topology is what defines the cardinal landmarks. The MRI microvolts of signal density are encoded as light and dark pixels. The skin can be represented as a surface of isovalues where the skin fat or water drops out into the low density air just above the skin. The SpiderWeb algorithm was developed to build isosurfaces for critical analysis and imaging. It is extensively described in Part 1 of this dissertation. The skin isosurface can also be used to register MRI coordinates by building a skin surface, displaying it on a high speed computer graphics workstation. We can explore this surface, and locate critical landmarks in much the same way as the neuromagnetometer operator visually locates cardinal landmarks with the Polhemus digitizing probe. The operator scans the skin surface by eye, looking for surface features that meet the criteria that define the landmarks. The Polhemus probe is used to record the position of a cardinal landmark on the skin. We can probe the skin isosurface in much the same way by means of a two dimensional cursor or “robot spider” that is constrained to remain on the SpiderWeb isosurface. The SpiderWeb surface is seeded with a moveable robot spider that creeps on the surface and can home in on curvature extremum [140] [141]. The bug can be positioned and steered about the surface by interactive operator intervention, or various navigation heuristics can be tested and the bug trajectory compared with hand estimation of the intuitively proper homing landmark criteria.

#### 5.3.1.6 Combining Metric Systems: The “CoordAligner”

The “CoordAligner” is a software package that identifies cardinal landmarks located by means the three techniques described above, namely Direct Localization on a slice, Pointers Technique, and Spider In Web Localization. The coordinate system

alias transformation matrix is calculated from the cardinal locations. No correction for measurement warpage error is made in the current implementation of the software.

Figure 5.19 shows an orthogonal system of coordinate axes in an machine dependant MRI coordinate system. In the CoordAligner software, cardinal landmarks are identified in space as small spheres. The cardinal landmarks are used to define the transformation matrix that will alias the MRI system coordinates into the PPN head-based coordinate system. After the PPN alias matrix is calculated, all of the coordinates in the MRI coordinate system are multiplied with the alias matrix. After transformation by the alias matrix, the cardinal landmarks are overlaid as best as the transform can, on each other. After this transformation a common set of numbers will specify the same location for both MRI and MSI coordinate systems. Figure 5.20 shows a typical overlaying of cardinal landmarks that results for such an alias transformation.

#### **5.3.1.7 Experimental Validation of the Techniques**

The experiments described below provide important validation of the obtainable accuracy and precision. The ultimate accuracy obtainable by the software is limited by the resolution of the Philips 1.5 T MRI system used in the present studies.

#### **5.3.2 Review of the State of the Art**

The generalized problem of coordinate system transformation has interesting parallels in the world of computer aided manufacturing, specifically in programming numerically controlled milling machines for metal working. Computer aided fixturing [179] [180] refers to various robotic techniques for reducing the work required to accurately and precisely position a casting or forging for metal working operations. Mishra [181] has solved a similar problem for transforming the coordinates of a Numerically Controlled (NC) milling machine to that of a casting or forging that is arbitrarily clamped (fixtured) to the machining table. Newer techniques of workpiece fixturing [182] [183] use an object based local coordinate system, similar to the head-based coordinate system idea that is developed here. Instead of precisely positioning the object to be machined in a fixture, the position and orientation of the rough object is determined by tracing the outline of the object to be machined with a mechanical robotic sensor. The poise of the casting is determined from its outline, and a local coordinate system,



based on the casting, can be calculated. The tool path coordinates are transformed, or aliased, into the local object-based coordinate system. Indeed, the problem of head registration can be recast as a problem of computer aided cephalic (head) “fixturing”.

Theoretical studies of the purely mechanics properties of registering two different medical image modalities were undertaken by Abche *et al.* Their first paper [184] measured the properties of test fiducial points in order to study how well the images from two different imaging modalities register on each other. Essentially, their procedure was to pretend that they had measured the positions of external markers from two medical imaging modalities. These positions involve measurement errors which are in general different in the two modalities. The errors were randomly introduced into the measurements of registration markers in various simulation computer runs. They then register the rigid bodies, assumed for the theoretical studies as spheres, by estimating the transformation parameters that minimize the  $\chi^2$  function of the distances associated with the marker positions. A numerical simplex or gradient algorithm is used to converge iteratively on a minimum  $\chi^2$  value. Surprisingly, they showed that the addition of more than 5 widely spaced markers did not significantly improve the residual error produced by the iterated solution parameters.

There is an large body of literature concerned with biological landmarks and deformations thereof. Bookstein reviewed the state of the art of landmark based metrics in [185] and [186]. He makes a distinction between landmark features defined by extremal points and landmarks as curvature maxima. As is pointed out in section 5.3.2.2, curvature-based definitions of landmarks depend on the feature having significant irregular, non-spherical, non-cylindrical, or “non-conic” geometric features [120]. A review of biological topology, with a new perspective on projections of biological shapes from higher dimensions is by Kergosien [187]. Measurements relying on identification of curvature extrema become less and less robust as the surface being probed for a landmark becomes more and more cylindrical or spherical. Automated analysis of surface features from the skin will become an increasingly important problem. This is especially true as the technology to capture skin surfaces via three dimensional light scanners matures. Cutting *et al.* discuss some of the first work on measuring and analysing skin surfaces as measured by a Cyberware Laser Light Scanner [188] [189]. Vannier [190] describes the modified Cendit facial surface range scanner. This system uses two high resolution video cameras to find facial fiducial points by Bhatia [191].

It has proven quite difficult to automatically locate skin landmark features. Early work with adapting stereophotogrammetry has been shown applicable



for the study of teeth, facial contours, body volume, and similar surface-based interpretations of biologic objects [192]. However, it is an immense effort to hand draw contour lines from planes of light illuminating a subject. Similarly, it would be even more cumbersome to measure skin landmarks with a caliper or a bulky articulated arm digitizer. Indeed there have been significant questions about the reliability of photogrammetry of the face [193]. New technology in surface digitization led by Cyberware Laboratories, Inc., [189] and Polhemus Navigation, Inc., [8] have made direct measurements from the skin surface practical for routine experimental and clinical use. Cutting [30] has argued for a skin based cephalometric coordinate system constructed from curvature extremum and curvature extrema as the landmarks themselves [28]. Recent work by Tirion *et al.* [173] and Gueziec *et al.* [194] and Monga *et al.* [195] are just beginning to realize the vision of Hilbert in 1932 [135] and Cutting in 1984 [29]. Early work on building databases of three dimensional landmarks for head X-ray (cephalogram) measurements was described in Grayson *et al.* [196] but has yet to become part of clinical orthodontic practice.

A new computer workstation for clinical orthodontics based on skin landmarks was announced as a commercial product of the Dolphin Imaging, Inc., in 1990 by Alexander and Chaconas *et al.* [197] [198]. It is based on a combination of a Lucite ear bars, a color video camera and associated digitizer hardware, and a sonic digitizer probe. The stated aim of the developers was to eliminate the need for cephalograms by accurate facial digitization via digitized video images and a sonic digitizing probe. Tooth positions are determined by dental casts mounted in the video camera field of view and registered by the sonic digitizer probe. Studies of its accuracy by Chaconas [199] showed consistently better accuracy in determining various facial landmarks with respect to cephalograms, and gave distance measures with standard deviations in the 1 mm range.

There has been a significant resistance to the incorporation of computer techniques in clinical image feature identification. The automated interpretation of facial features will come in the future, but for the present, interactive and computer aided interactive techniques are being developed to interpret and digitize image and volume data. There is evidence that the interpretation of human facial features is handled by a special area of the brain [85] [200] [201]. The existence of special regions of the brain devoted to facial recognition is an indication of the difficulty in recognizing, locating, and computing facial landmarks. However, we show that an experienced operator using a Polhemus digitizer probe can reliably and repeatedly sight landmarks within 1.2 mm, as shown in this dissertation. However, developing computer algorithms to automate this process has proven quite difficult.

### 5.3.2.1 The Direct Method for Finding Landmarks

The most popular approach to relate external landmarks to features imaged within the head is to place an imagable marker on each landmark. In our experience, it is difficult and time consuming to precisely orient and position tomographic slices to pass directly through a marker to obtain accurate locations of the landmark. It is possible to obtain estimates of capsule position, but the capsule location is not congruent with the landmark location. For instance, in MRI recordings a Vitamin E capsule is often used. As the capsule may be larger than the typical slice thickness ( $> 3 \text{ mm}$ ), the location of the slice with respect to the center of the marker or the location where the object touches the skin, is unknown. Pohjonen *et al.* [202] describe a software system based on using Vitamin E capsules situated in the ear canals, one on nasion, and yet another between the teeth. The accuracy in determining such locations very much depends on careful interpretation of the marker's image. The Pointer technique avoids any need for precise alignment of recording slices, and it provides an accurate indication of the landmark where as it touches the skin. Thus the information in the image precisely defines a location on the scalp, not just the location of some feature of the marker. The "Best Line and Point" algorithm and program was developed to accurately locate cardinal landmarks, which may lie at the edge or between slices, in a stack of image slices. This method permits localization of a landmark in stacks of arbitrary position or orientation.

### 5.3.2.2 The Hat of Pelizzari

Another method for relating images of different modalities to establish a coordinate system registration is to match homologous objects by their outline contours. Pelizzari [203] did this with flexible tubes of imagable material made into a mesh to cover the scalp. In this way, an outline of the surface of the head can be recorded by CT, MRI, and PET, by generating a surface "hat" from the imagable traces of the hat mesh.

This technique assumes that given two 3 dimensional models of an object, a unique coordinate transformation can be found which, when applied to one of the models, makes the two fit together with a minimal error distance and volume between the two surfaces. This is analogous to finding the optimum placement of a custom fitted rigid hat on a head. The operator traces contour rings, hat markers, and points in slices of the head. The collection of points represents the skin of the scalp. A set of points, and rings representing the position of the head in each modality to be registered, such as CT, MRI, and PET, is recorded, and each is projected into one



modality by a transformation. The allowed class of transformations includes rigid body rotations and translation (poise as specified by three Euler angles, and three orthogonal translation displacements) and linear scaling of the “hat” along three orthogonal axes.

The progress of the optimization of translation and rotation parameters search is assisted interactively by the operator. Operator interaction reduces the time required for the Powell’s numerical algorithm to converge to a minimal error volume hat position and orientation. When scale transformations are permitted, the residual homologous distance error can be made as small as desired. In addition to interactive parameter guidance, Powell’s numerical iteration method [204] is used to calculate the transformation matrix that minimizes the distance between contour lines of different modalities of the same subjects. Pelizzari [203] notes :

For a highly symmetric object such as a sphere or cylinder, clearly a unique transformation could not be found based on only matching surfaces. Fortunately, such perfect symmetry seems sufficiently lacking in human heads that in all cases so far processed ( $\sim 50$  as of this writing) the method has succeeded.

This Hat method depends on small irregularities and subtle shape features present on the otherwise featureless upper head. The reported accuracy are at the limit of the imaging modality, as tested on approximately 50 subjects [203]. Three dimensional accuracy of 1 to 2 pixels of approximately 1 mm for CT and MRI were obtained. Tests with phantoms indicated a residual RMS error distances of 0.73 mm for CT/CT registration, 1.16 mm for CT/MRI, and 1.83 mm for MR/PET.

In our opinion, the effort required to form the mesh and ensure it touches the scalp everywhere provides no better results than the much simpler Pointer technique.

Allan Evans *et al.* [205] at the Neuro-Imaging Laboratory of McGill University have extended the hat concept for use in building a brain atlas for relating Positron Emission Tomography (PET) functional information to CT and MRI anatomic maps.

PET images offer important functional information but imprecise anatomical reference geometric information because of poor spatial resolution and because



the radioactive tracer distribution does not adequately penetrate and reflect underlying anatomical variation. These problems are often exacerbated when imaging a pathological brain, where extensive disruption of normal functional patterns can occur. Hence, accurate and reproducible analysis of different types of PET images requires additional information from the structural imaging modalities of Magnetic Resonance Imaging (MRI) or Computed Tomography (CT).

Evans *et al.* [205] have developed procedures for the registration of MRI and PET image planes using fiducial markers in the patient setup. They also implemented a method for transferring the MRI anatomical information to the PET analysis via a computerized brain atlas that is modified, interactively and tediously, plane by plane, to fit each subject's brain. They describe their implementation as two major steps in correlative analysis: (i) registration of anatomical and functional image volume by application of an algorithm to minimize the mean-square distance between equivalent points in the two volumes, and (ii) the segmentation of the image volumes by means of an interactively adjustable three dimensional Volume-Of-Interest (VOI) atlas derived from contiguous 2 mm thick MRI images planes. Evans *et al.* [205] go on to describe the various techniques of image warpage that are based on the Procrustes Method [206] [207] [208].

### 5.3.2.3 Qsh

Qsh is a general purpose medical imaging viewing, analysis, and cross-modality registration tool developed and written by Maguire and Noz, and is described in Noz *et al.* 1988[209]. It is used at New York University Medical Center by a number of researchers [53]. It is also used as the foundation for the CoordAligner and the SpiderWeb system. Essentially, this algorithm re-slices volume pixel data from each modality on a 128 by 128 pixel format, and then matches them within the re-sampled slice plane with trilinear and other interpolation functions. The slices are adjusted within the plane by rotation, translation, scale, and skew interactive adjustments available in Qsh [137][210]. A main difference between Qsh and the CoordAligner, described in this dissertation, is that the CoordAligner does not move any data *in situ*. It calculates an alias transform matrix for each stack and composite (system of stacks or other objects) instead of resampling data.

Qsh landmarks are determined by comparing two homologous slice images side by side on the computer graphics workstation screen. The registration transformations are made on the slice to be matched relative to the reference slice. The skilled

radiologist or neuroanatomist operator normally chooses 7 to 12 pairs of landmarks. The landmarks pairs from the reference image is cross-correlated with each of the points in a neighborhood around the corresponding landmarks from the other image. The cross-correlation function chooses the pair exhibiting the strongest correlation and updates the coordinates on the image to be registered. The end result of the correlation function is that it produces sets of coordinate pairs that relate the respective locations of the anatomical landmarks in the homologous reference slice and the slice that is registered.

Essentially, a cross correlation function is generated that is used by a polynomial transform function to calculate the alias registration matrix. The Gauss-Jordan inversion of the algorithm is used to obtain the eigenvalues of the matrix of coefficients. These eigenvalues together with a resampling technique determine the new values for each pixel in the image to be moved. The images are required to have the same size.

This technique was validated with a Jaszczak phantom [211]. This is a clear plastic cylinder containing a set of shapes for determining spatial resolution. The user can place plastic spheres of different sizes on rods of different lengths and fix them to various known sites inside the phantom. The plastic spheres are filled with a contrast agent appropriate to the imaging modality to be tested. For SPECT, the cylinders and spheres were filled with water and 925 MBq (25mCi) of the radionuclide Technetium-99m ( $^{99m}\text{Tc}$ ) Pertechnetate. Surrounding that were affixed a ring of 5.55 MBq Iodine-123  $^{123}\text{I}$  pills around the outside of the cylinder at the level of the Technetium-99m spheres. CT images were made of the phantom filled with iodinated contrast material and water. MR contrast was obtained with a ring of Vitamin E capsules to the outside of the cylinder at the level of the spheres. The results were valid to within one to two pixels, or 3 mm to 6 mm. Because the spatial resolution of the SPECT is about 1.2 cm, further analysis of error below this resolution is not significant. Extensive experimental and theoretical analysis of error in resampled slices with different slice orientations was examined by Maguire in [212]. The main problem cited by Maguire *et al.* [137] is the extensive user interaction required:

All of the methods currently successful for registering images from the same or different modalities require extensive interaction by the radiologist, anatomist, or other trained observer and require substantial processing time. Our current method requires operator time for the oblique reconstruction and for selection of suitable landmarks. In addition, if we wished to use the values of the transformed pixels, we would need to do a more



time-consuming reprojection, registration, and resampling. Our goal is to investigate methods that decrease the user interaction required and improve processing time.

We have demonstrated that the Pointer technique overcomes these problems, and that the CoordAligner is a fast and simple method to register disparate data sets without extensive unreproducible user interaction.

#### **5.3.2.4 Longitudinal Fissure as a Landmark**

Internal landmark features in the brain can be used to establish a coordinate system. If a prominent anatomic structure can be identified, and it defines a plane, it can be used to define a coordinate system around that structure. Brummer [213] uses the plane of the longitudinal fissure as calculated from the Hough Transform to establish a coordinate system without using cardinal landmarks. An interesting property of this coordinate system is that it precisely divides the left and right hemispheres. This technique should prove valuable in functional laterality studies with MEG [81]. This is because the sign of the  $Y$  coordinate value will define which hemisphere is being mapped, or is functionally active, for all values of  $Z$  or  $X$ .

The longitudinal fissure is the gap between the two hemispheres in the brain. This fissure is filled with cerebro-spinal fluid (CSF). It is a fairly well defined planar structure in most human brains. Volumetric analysis, for example, requires the segmentation of left and right brain volumes. The left and right boundaries of the brain hemispheres are defined by the location of the longitudinal fissure. Apart from being a useful feature in its own right for image analysis, the plane of the longitudinal fissure can be used as a landmark in establishing a scanning device-independent coordinate system of the head. A Hough transform of the MRI image of the region including the longitudinal fissure is used to calculate the coordinate system basis vector directions and the origin location. Once the plane of the longitudinal fissure is determined, there is only one degree of freedom to be resolved on that plane. Any of a number of internal or external landmarks can be used for this purpose. Generally, the skin surface provides a suitable fixation.

The Hough transform was proposed in 1962 by Hough in a U.S. Patent [214]. The technique was optimized by Duda and Hart [215] for detection of straight lines. It is usually used to transform straight lines to points on the Hough space. This



idea can be extended to three dimensions, where it will transform an almost planar feature in three dimension cartesian space to a pointed unimodal spread region in three dimensional Hough space. To understand the Hough transform idea, consider a location  $(x, y)$  in a cartesian coordinate system in a 2d plane. For any point  $P(x, y)$ , we can define a line  $l$  through  $P$ , with normal vector  $\vec{n}_1$  :

$$(5.6) \quad \vec{n}_1 = \begin{bmatrix} -\sin \omega \\ \cos \omega \end{bmatrix}$$

Where  $\omega$  is the intersection angle of the line with the  $x$  axis.

$$(5.7) \quad \vec{P} \cdot \vec{n}_1 - d = 0$$

This equation when solved for  $d$  gives a mapping for any line in the cartesian coordinate space to a point at  $(\omega, d)$  in Hough space. Each value of  $\omega$  in the range  $[0, 2\pi]$  defines a unique value of  $d$ . In the  $(\omega, d)$  Hough parameter space. A point  $(x, y)$  in the plane appears as a sinusoid, while a line is characterized by a single point  $(\omega, d)$ . Similarly, we can see how any planar formation in space can map to a point in a generalized three dimensional Hough transform.

Because medical images are noisy and biologic shapes refuse to conform to planes, spheres, and other conic shapes, a plane like formation in the brain will cluster about a point in Hough space. The center of the blobby cluster represents the best fitting plane of the longitudinal fissure.

The plane found at the center of a cluster of Hough points can be used to define a coordinate system. The origin of the coordinate system can be fixed on the plane by finding the center of gravity of a resampled congruent slice. This is a precisely midsagittal slice. The intersection of the midsagittal plane and the highest point of the skin vertically, at the apex of the head, can define another point to fix the position of the emerging coordinate triad. Alternatively, we can find the most anterior intersection of the nose and the midsagittal plane and use that to define the  $X$  axis fixation point.

A complete coordinate system can be constructed in this way. A Hough transform, in this setting, finds the normal vector of the midsagittal plane. The center of the midsagittal plane defines the origin. The most cephalic or rostral intersection of the skin with the midsagittal plane defines the  $Z$  or the  $X$  axis vector respectively. The normal vector of the plane, fixed at the origin, defines the  $Y$  axis. The advantage of the midsagittal plane and the more cephalad intersection point is that there are no

features at the cephalic apex of the head. A particular importance of this coordinate system is that a  $+Y$  value for any  $Z$  or  $X$  is in left hemisphere, and a  $-Y$  value is similarly in the right hemisphere. In the alphon studies described in Section 9, and in any brain dominance study, this certain knowledge is critical to separating closely spaced sources. A similar use of least squares cluster analysis and the Hough transform was presented by Chen *et al.* [216]. However, it is practically impossible for the neuromagnetometer operator to visualize the midsagittal plane to register the machine coordinate system fixation. The main value of internal feature registration algorithms is to re-orient the coordinate system of the head for functional studies after the external, registration is obtained.

Fox *et al.* [217] introduced a method to register PET images by determining the positions of the anterior and posterior commissures with a sagittal X-ray image obtained with the subject in the same position. However, methods of this kind are unsuitable for the purpose of obtaining a critical magnetic functional and MRI anatomical registration, since they do not relate to external landmarks that are most convenient for MSI.

### 5.3.3 Other Registration systems

There are non-landmark based surface registration methods. Given that measurements of cardinal landmarks for head registration are necessarily skin based, we will look briefly at methods that propound to not need the identification of point landmarks.

The ANALYZE<sup>TM</sup> package was developed by Robb *et al.* [218] [219] [220] [221] [222] [223] for quantitative analysis of biomedical images from a number of disparate medical imaging technologies. A registration facility for multimodality medical image registration by surface matching was presented by Jiang, Robb, and Holton [224]. Their approach to registration of multiple images has three steps. The first is defining corresponding features between the different image data sets, finding the matching transformation, and transforming one or more of the images to bring them into spatial registration with respect to each others features.

The Jiang, Robb, and Holton [224] approach to three dimensional image registration is based on “chamfer” matching. In this context, a chamfer is a smoothed edge feature that is fitted between two objects. This approach combines a measure of the registration cost applicable to arbitrarily complex multidimensional surfaces



using a multi-scale multiple starting seed point search strategy. This strategy greatly increases the probability of efficiently locating the global minimum without manual intervention. Although their initial results with this algorithm were promising, they admit it requires additional refinement, comprehensive simulations, and phantom-based testing for complete validation. The problem as to which surfaces are best to match with different imaging modalities must also be addressed.

Essentially, this method starts with identifying two volume images to be registered, one as the base image and the other as the image to match. The “match” image is subsequently transformed to align with the base image. Object contours are extracted from both the base and match images by using semiautomatic image segmentations available as part of the ANALYZE<sup>TM</sup> software package. The contours from these base images are stacked together to form three dimensional surfaces. A limited number of points are uniformly sampled from the base and match contour data sets, and only these points are used for the registration. Chamfer matching is accomplished by performing a distance transformation which converts a binary surface into a grey-level image in which all voxels are assigned a value corresponding to the distance to the nearest surface. Robust matching is assured if there is no rotational symmetry between the two surfaces, and that there is no warpage or stretching between the base and match surface.

The Jiang, Robb, and Holton [224] results are presented for image sets of the brain and the hip. The modalities registered were: MRI *vs.* PET, MRI *vs.* SPECT; MRI *vs.* CT; and Serial T1 MRI *vs.* T2 MRI studies taken several months apart. The algorithms were evaluated by simulation registration studies. For the simulation purposes, the same object was used for both base and match registration. The image size was  $256 \times 256 \times 256$  pixels and the size of each voxel was  $0.938 \times 0.938 \times 0.938$  mm. To simulate non-cubic voxel data, the base surface was resampled at every third slice and the match surface was resampled at every fifth slice after arbitrary geometric transformation. Gaussian noise with zero mean and 5 % standard deviation, equivalent to 7.0 pixels, 6.6 mm thickness, was added to the surface coordinates to roughen the surfaces. Simulated registration studies were reported with three error statistics, namely the surface error, the centroid error and the object error. The surface error is the average distance of all the match surface points. The surface error was reported at 0.28 cm. A centroid error is the difference in position between the registered object’s centroid. They report a value of 0.17 cm. An object error is the average distance from the same number of points as the surface error, but the point pairs are randomly rearranged about the reference and match object surface. The reported object error was 0.21 cm. The complete error was characterized by giving the residual transformation matrix. If no further registration was possible,



the residual matrix would be an identity matrix. The difference between this matrix and the identity matrix represents the residual unregistrable errors between the two objects. They present the mean and standard deviation matrix of this matrix for ten cases. However, it is not clear what these figures represent, except that the values are all extremely small, and less than the pixel sizes of the source data. They did not do any skin surface matching. This paper had the most interesting technique in the literature surveyed specifying registration error in the image.

Friets [225] has developed a “frameless” stereotactic operating microscope for neurosurgery which required merging of the CT coordinate frame with the view through the eyepiece. The location of the microscope focal plane is fixed via ultrasound digitizer transducers. This gives the location of the microscope with respect to tattooed landmarks on the patient’s skin. These landmarks are recorded in CT images by placing glass beads on them. The reported precision and accuracy for registration of the CT and microscope is 0.8 mm.

Wieringa and Peters [226] at the University of Twente, the Netherlands, discuss the use of a “V-block” registration device. A Plexiglas wedge with two tubes containing a suitable contrast agent is placed on the head over the cardinal landmarks. While it is clear from the figures that the centerlines of the contrast agent tubes are designed to converge at a point, no mention of any software or algorithm to calculate this point of convergence from MRI slices is mentioned in the preprint.

Perhaps the most visually interesting image registration technique involves registering ultrasound images and the view through the eye pieces of a heads up display. Bajura and Fuchs, *et al.* [139] described a prototype ultrasound system where the user wears a modified head mounted display. A head mounted display is a type of helmet with two small video viewfinder displays that fill the user’s field of view. They are modified so that the user sees through half silvered mirrors, reflecting the video for each eye, and transmitting the view from the goggles. In the way of a “heads-up display”, the three dimensional ultrasound information is shown as it would appear from the viewpoint of the operator. This requires simultaneously tracking the head of the user, and the position of the ultrasound probe. The image of what the ultrasound sees inside the patient is registered from the point of view of the operators head mounted display. A technique of arbitrary plane three dimensional reconstruction and registration is described in King *et al.* [227], and Ohbuchi [75] describes an incremental volume reconstruction and rendering system, but real time 3d reconstruction is not possible with the current generation of computer graphics workstations. Experiments were conducted in real time 3D visualization of an *in vivo*

fetus *in situ* with a pregnant woman. These studies were not quantitative in nature, and no figures for the accuracy or precision of the registration were reported.





## 5.4 Cardinal Landmarks

Implicit in the selection of a set of useful landmarks is that they can be easily and correctly identified. Moreover, the landmark feature must be easily accessible to a pointing device that can accurately record its position (*e.g.*, by using a Polhemus 3Space<sup>TM</sup> [8] three dimensional digitizing probe). Additionally, the landmarks should be widely spaced so as to provide maximum tolerance of errors in locating them. It is important in locating landmarks with respect to easily seen features of the head that they have a precise anatomic definition. For instance, the nasion could be defined as “the saddle point of the nose where it meets the forehead”. Also, the left and right preauricular points could be defined as the location where the ridge of the anterior tragus (external ear flap anterior to the ear canal) merges with the scalp on its upward (vertex) end. This differs somewhat from the less precise definition on which the ten–twenty system for placing EEG electrodes is based. In that definition, the periauricular points are where the anterior tragus (external ear flaps) merge with the scalp on the face side [176] [228].

For our purposes, we assume that the landmarks are invariant in the time scale of our studies. However, should longitudinal studies be attempted of brain function in a growing child, then the issues of growth will have to be confronted [229]. Once landmarks are identified, the next problem is to compare one constellation of landmark points, space curves, and surfaces with respect to a reference landmark constellation. The class of techniques used for translating, rotating, and warping one landmark constellation into another is called the “Procrustes Method” [207] [208] [230]. The ancient Greek innkeeper, Procrustes, had only one bed for all his guests. He was renowned for chopping the legs of guests who were too tall, and stretching those who were too short [231]. The modern Procrustes problem in our context is how to fit the head measured with Magnetic Resonance Imaging into the data digitized by an experienced operator with the Polhemus probe as part of a neuromagnetometer recording study. Essentially, Procrustes concerns itself with the “optimal” superimposition of two objects, and the distribution of the residual error. Mathematical techniques include: least–squares; resistant fits (where a few homologous landmarks are wildly different); affine transformations; and thin-plate spline for energy minimization in warping the two objects to fit together [232]. However, for our purposes, we show in this study that the transformation of each separate data set into the PPN head–based coordinate system is more than satisfactory. This is because we show that the residual error in the registration, as measured by the methods discussed below, is less than the finest resolution measuring modality. In this study the hand digitized cardinal landmarks locations had the highest precision of all the measurements

obtained.

The following operational definition has been adopted which includes both curvature maxima and location extremum. A cardinal landmark is precisely defined by the skin topography. The following operational definition has been adopted which includes both curvature maxima and location extremum. A cardinal landmark is precisely defined by the skin topography.

For the nasion, it is the most posterior point on the crest ridge profile of the nose at the saddle point where the bridge of the nose meets the forehead. For the periauricular points it is the most medial point on the superior portion of the crest ridge of the anterior tragus.

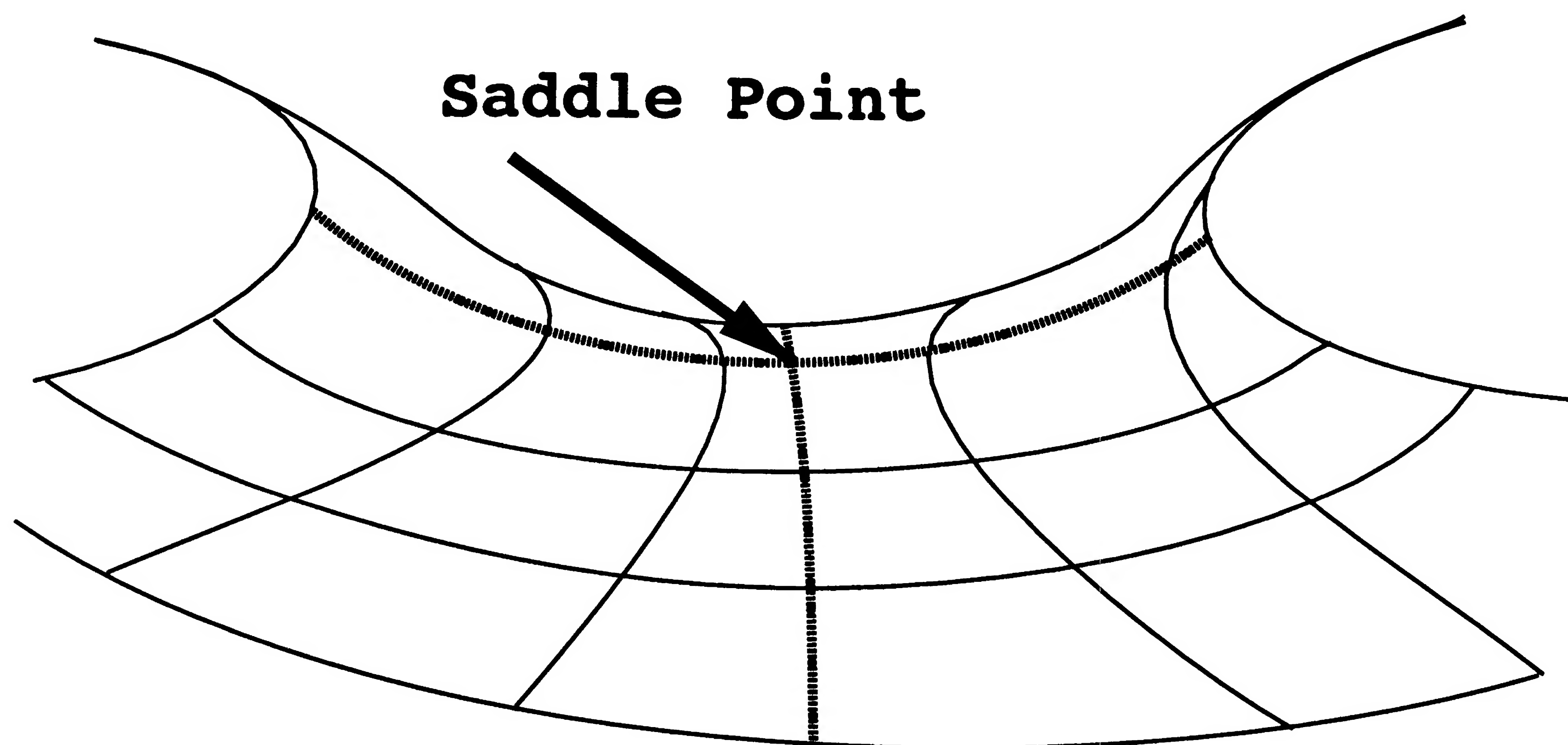


Figure 5.5: A landmark as the lowest point on the ridge profile at the saddle

These points have been chosen to define the “PPN head-based coordinate system” [151] [233] on which the Pointer technique is based. When used for clinical applications, or for other applications where maximum accuracy over many days is desired, the cardinal landmarks can be precisely established by enduring ink marks applied to the skin. In addition, the particular locations on an individual are documented in photographs. This reduces inter-operator variability, and improves precision of measurements taken over a long term. We show in Table 6.2 and 6.3 that the measurements of cardinal landmarks locations are extremely accurate and

precise. The precision of repeated measures is less than 0.4 mm standard deviation by an experienced operator.



#### **5.4.0.1 The classical anatomic orientations: axial, coronal, and sagittal**

The data sets from each MRI study are in the form of a series of 2D slices in the form of a stack. The slices within each stack are oriented in three cardinal directions, based on the subjects anatomical orientation. A stack of slices, or any individual slice in a MRI system can have an arbitrary orientation. For conventional MRI studies, the basic three anatomic orientations are sagittal, axial, and coronal[234][235].

##### **Sagittal**

The midsagittal plane is the midline plane dividing the left and right hemisphere of the head. A sagittally oriented slice is parallel to the sagittal plane, and often has a constant  $y$  value across the surface of the plane. A sagittal stack starts slicing through the head from the right ear, through the midsagittal plane, and ends up slicing through the left ear.

##### **Axial**

The axial orientation is parallel to the spinal column. An axial plane generally has a constant  $z$  value across the surface of the plane. An axial stack starts slicing the head about the base of the neck, and moves headward, until it is slicing through the apex of the head, at the top of the scalp.

##### **Coronal**

The coronal orientation is in the direction of the nose, anterior, or rostrum. A coronal plane generally has a constant  $x$  value across the surface of the plane. A coronal stack of slices starts at the occipital pole of the head, the posterior pole of the head, and slices the head until it is just slicing the tip of the nose.

#### **5.4.1 The PPN Transform Algorithm**

A minimum of three non-colinear cardinal landmarks defines a basis for a head-based coordinate system. Coordinate systems can be generated with more than three cardinal landmarks so that overdetermined solution is achieved. This provides an internal consistency check. There are two kinds of algorithms that can be used to produce a matrix to relate one coordinate system into another. The first kind of algorithm attempts to find the best fit of one constellation of landmarks into a homologous landmark constellation. It essentially fits one imaging modality into another. The second

type fits both modalities into a different, and common coordinate system. In this case an alias matrix transforms the coordinate system (but not the actual landmark locations) of the MEG machine coordinates and the MRI machine coordinate system coordinates into the common PPN head-frame coordinates. The CoordAligner software is capable of both alignment schemes, but for these studies, all coordinates were transformed into the common, PPN head-frame coordinate system. One coordinate system was not merged into another, but both coordinates systems were transformed into the PPN head-frame.

A numerical approach to fitting two coordinate systems together uses the Amoeba algorithm to calculate the parameters of a rigid body transformation [236][237][204]. This is a type of constrained Procrustes transformation. The constraint is that the transformation must be orthonormal, and no scale changes are introduced. Basically, this solves the transformation of rotation and translation parameters to find a minima in the distances between homologous cardinal landmarks. This approach was originally used by Cutting [28], but newer methods have been adopted [30]. This has the advantage of allowing more than three basis-defining cardinal points, but it has numerical stability problems in that the Amoeba, using only local information, can frequently get trapped in a local minimum. Another generalized multi-parameter numerical approach is Powell's method [204], and it had been used by Cutting [30] and Pelizzari [203], as described in Section 5.3.2.2.

The PPN transform [151] constructs an orthogonal axis triad from three basis vectors constructed from the cardinal landmarks. It is similar to the classic Gram-Schmidt orthonormalization problem [238] [239]. Because the two periauricular points and nasion are used to define this system, we build an orthonormal basis from these points. This basis is called the PPN Headframe. It is necessary and sufficient to use three non-collinear cardinal points. The PPN transform does not distribute the residual error evenly, as will be described later. An orthogonal basis is constructed from the cardinal landmarks and they serve as the  $X$ ,  $Y$ , and  $Z$  axes.

- 1 First construct line 1 from the right to the left periauricular point.
- 2 Then calculate the midpoint of line 1. That is the origin of the PPN Head-frame.
- 3 A line is erected from the origin out through nasion. That line is the  $X$  axis.
- 4 A vector is constructed by the vector cross product of the  $X$  axis and line 1. This vector is fixed to pass through the origin. This line is the  $Z$  axis. It passes through the top of the head.



- 5 A vector perpendicular to the Z axis and the X axis is constructed by taking the cross product with the Z axis and the X axis. This vector is fixed to pass through the origin. This is the Y axis.

This PPN Headframe serves as the common coordinate system for MSI and MRI. The coordinate system used in MSI is a right-handed coordinate system. The  $+y$  direction is toward the left hand of the subject. In MRI, the conventional radiological coordinate system is left-handed. The  $+y$  direction is toward the right hand of the subject.

The PPN Headframe coordinate system is spanned by three axes,  $x, y, z$ . The origin lies midway between the left and right periauricular points. This places the origin close to, but is not congruent to, the center of mass (centroid) of the head. The properties and construction of the axes is described next.

- X** The positive  $x$  axis of the system is the line from the origin that passes out of the head through the nasion. Therefore, the positive  $x$  axis passes from the origin through the front of the head. This approximately corresponds to the Anterior–Posterior (AP) axis in MRI terminology. The AP axis increases in the rostral or anterior direction.
- Z** The positive  $z$  axis is the line from the origin that passes through the top of the head. It is perpendicular to both the  $x$  axis and the line between periauricular points. It is perpendicular to the plane these two axes define. It emerges from the head near the vertex, but generally not exactly through it. This corresponds approximately to the Cranial–Caudal (CC) axis in MRI terminology. The CC axis increases in the cranial or headward direction.
- Y** The positive  $y$  axis is the line from the origin that passes through the left side of the head. It is perpendicular to the  $x$  axis and the  $z$  axis. This axis is opposite in direction from the MRI Left–Right (LR) axis. The  $y$  axis need not exactly coincide with the line between the periauricular points: it will pass out through the left side of the head anterior to the left periauricular point if the left ear lies posterior to right ear [240]. This axis is approximately opposite in the direction from the MRI Left–Right (LR) axis.

The PPN Headframe is a right hand coordinate system; the MRI metric is left-handed. The  $+y$  direction is to the subject's left in PPN, while the  $+y$  direction is to the clinical observer's left, with the clinical observer facing the patient face to face.



### 5.4.2 Computing the PPN Headframe Axes

Mathematically, construction of axes for the PPN Headframe can be reduced to finding the point that we will call the origin, rotating a set of orthogonal axes, and then translating that point to 0, 0, 0 in the rotated coordinate system.

$$(5.8) \quad \vec{O} = \begin{vmatrix} X_{origin} \\ Y_{origin} \\ Z_{origin} \end{vmatrix} = \begin{vmatrix} (X_{LeftPA} - X_{RightPA})/2 \\ (Y_{LeftPA} - Y_{RightPA})/2 \\ (Z_{LeftPA} - Z_{RightPA})/2 \end{vmatrix}$$

The  $+x$  direction is constructed by computing the vector  $\vec{x}$  from  $\{X, Y, Z\}_{origin}$  to the  $\{X, Y, Z\}_{Nasion}$ . The vector  $\vec{z}$  is constructed by taking the cross product of  $\vec{x}$  and the vector constructed from Left PA and the Right PA:

$$(5.9) \quad \vec{X} = | \{ (X_{Nasion} - X_{origin}), (Y_{Nasion} - Y_{origin}), (Z_{Nasion} - Z_{origin}) \} |$$

$$(5.10) \quad \vec{Z} = | \vec{X} \times \{ (X_{LeftPA} - X_{RightPA}), (Y_{LeftPA} - Y_{RightPA}), (Z_{LeftPA} - Z_{RightPA}) \} |$$

$$(5.11) \quad \vec{Y} = | \vec{X} \times \vec{Z} |$$

The set of normalized vectors represent the orthonormal rotation required to transform the values in the original coordinate system into the PPN Headframe.

$$(5.12) \quad \mathbf{R} = \begin{vmatrix} \vec{X} \\ \vec{Y} \\ \vec{Z} \end{vmatrix}^T$$

The translation is specified in the rotated coordinate system, so we dot the translation vector into the PPN system.

$$(5.13) \quad \vec{T} = \mathbf{R} \cdot \vec{O}$$

Therefore, the complete transformation matrix to take a value in the MRI coordinate system to the PPN system is given by:

$$(5.14) \quad \mathbf{A} = \begin{vmatrix} 0 \\ |\mathbf{R}| & 0 \\ 0 \\ \vec{T} & 1 \end{vmatrix} = \begin{vmatrix} Rx_1 & Ry_1 & Rz_1 & 0 \\ Rx_2 & Ry_2 & Rz_2 & 0 \\ Rx_3 & Ry_3 & Rz_3 & 0 \\ T_x & T_y & T_z & 1 \end{vmatrix}$$

To obtain the value of a coordinate in the PPN Headframe from a value in the MRI coordinate system, we dot the value  $X$  with  $\mathbf{A}$ .

$$(5.15) \quad \mathbf{X}' = \mathbf{X} \cdot |\mathbf{A}|$$

Source code in C to generate a PPN Headframe transform matrix is given in the Appendix A.1.

There are many other methods to produce a coordinate system for the head. As is standard engineering practice, many problems are uniquely suited to simple solution in a particular coordinate system. There is no requirement to find the “only” or “best” coordinate system, once an accurate and precise registration has been obtained. The coordinate system should simplify the particular problem we wish to solve. Indeed, a simple conversion from cartesian coordinate systems to a polar coordinate system has been shown by Alakuijala to make segmentation of brain images simpler [241]. Brummer [213] argues for the use of the Hough transform to find the longitudinal fissure and build a coordinate system of the head.

In the traditional stereotactic head frame, the  $y$  axis is the center of the ear rod passing through the auditory meatus (ear canal hole). The main advantage of this technique is that the head can be held rigid by means of ear rods [202]. However, clearly this is unsuitable for auditory stimulation experiments. Additionally, normal subjects will not tolerate ear rods. On the other hand, the Pointer technique does

not require finding the center of the ear canal, and does not require an uncomfortable head frame and ear bars.

A feature of the PPN construction is that it is insensitive to errors in certain directions. The nasion direction is not affected by errors in the  $X$  direction. It is only sensitive to errors in the  $Z$  or  $Y$  direction. Similarly, the periauricular points are located, the digitizer stylus can be pressed into the skin normal direction without causing errors. Cardinal landmarks expressed in the PPN Headframe, have these properties:

- The  $Z$  coordinate values for all of the cardinal points are 0.0, or the cardinal landmarks are all on the  $Z = 0$  plane.
- The  $Y$  value for the nasion is zero.
- The  $Y$  coordinates for the left and right periauricular points have the same magnitude, but opposite signs.
- The right periauricular point  $Y$  value is negative, and the left  $Y$  periauricular value is positive.
- The  $X$  coordinate of LPA and RPA are the same, but opposite in sign.

#### 5.4.2.1 Error Analysis of PPN Cardinal Landmarks

Estimating the error in the constructed PPN coordinate system from MRI data is done by comparing the positions of the cardinal landmarks. One set of landmark measurements was obtained from a MRI stack with their positions as recorded in the PPN Headframe and the reference measurements from the Polhemus digitizer probe directly on the subject's skin. We choose three criteria to evaluate the quality of the constructed PPN Headframe:

- The **Slop Error** distance is the sum of distances between each pair of homologous landmarks. A pair of homologous landmarks is one landmark as determined with the digitizer probe on the skin and the same, or homologous landmark as measured from the MRI stack. A perfectly congruent coordinate system registration would have zero slop. Figure 5.6 illustrates slop between two homologous triangles.



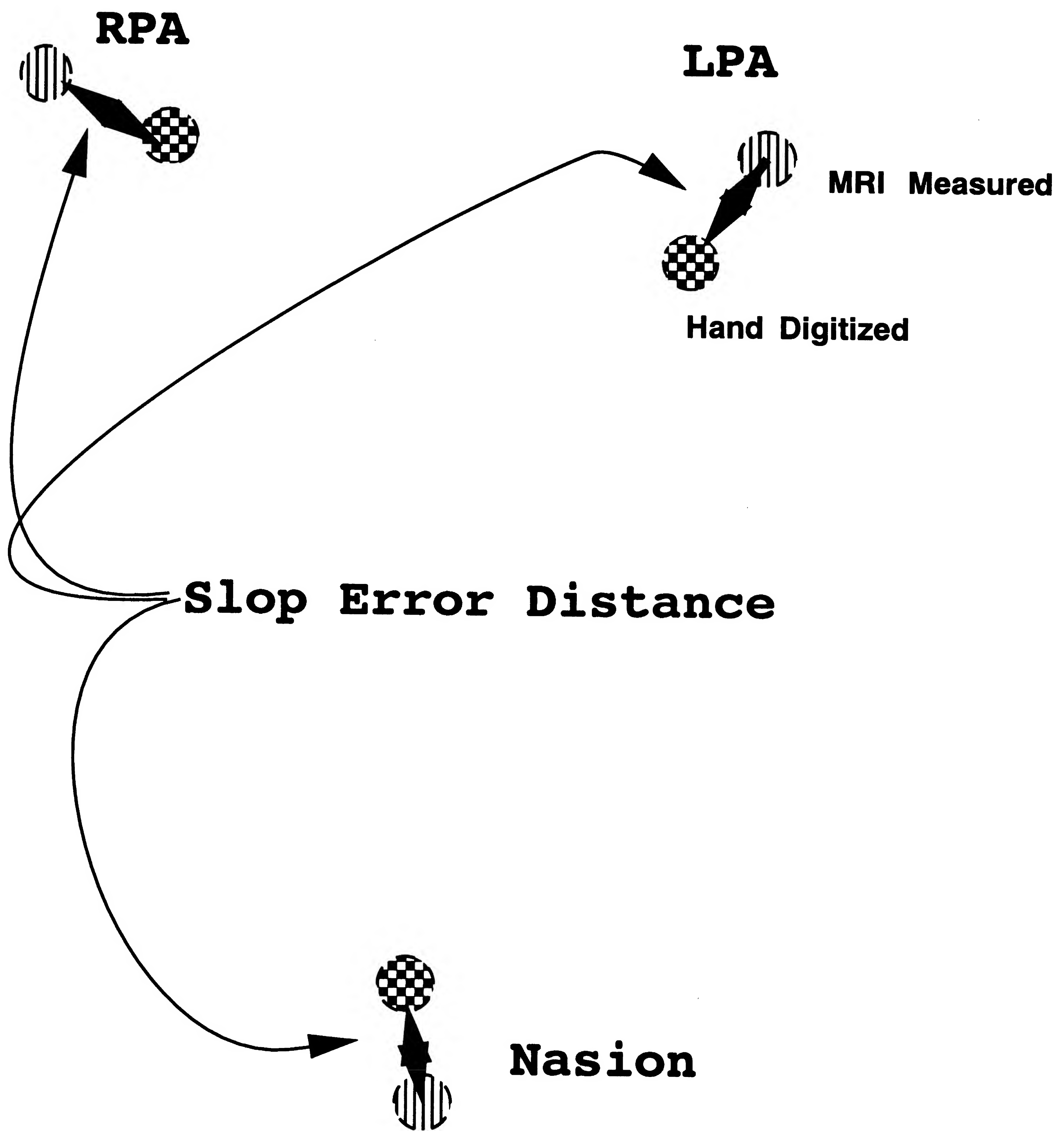


Figure 5.6: Slop Distance Error.

- The **Warpage Error Percentage** measures discrepancy in distance between each homologous pair of landmarks. Figure 5.7 illustrates warpage error. A homologous pair is the same landmark measured by different techniques. We compare the distances measured with the Polhemus digitizing probe and the distances measured in the MRI stacks. If there were no error, the distances would be identical. However, there are errors that appear as different distances as measured by different systems. The base of the percentage is the distance directly measured with the digitizer stylus point. Zero percent warpage means that there is no stretch or contraction in the distance between homologous pairs of the landmarks. A percent warpage is calculated for each leg pair of the each coordinate system landmark triad. The averaged percentage warpage provides a figure of merit for the accuracy of the how the two coordinate systems measure the same distances. One system can be larger or smaller with respect to the other. For example, a positive warpage means that one system is measuring consistently larger than the reference system.
- The **Wobble Error Distance** is the difference in the distance between the X value of the left or right Periauricular points. Figure 5.8 illustrates the Wobble error distance. Recall that the x value is the same, but opposite in sign, so either figure will serve the same purpose. This difference in values is a measure of the asymmetry of the ears. The difference in the difference values is an figure of merit of how well the coordinate systems are measuring the same asymmetry in the different measurement modalities.

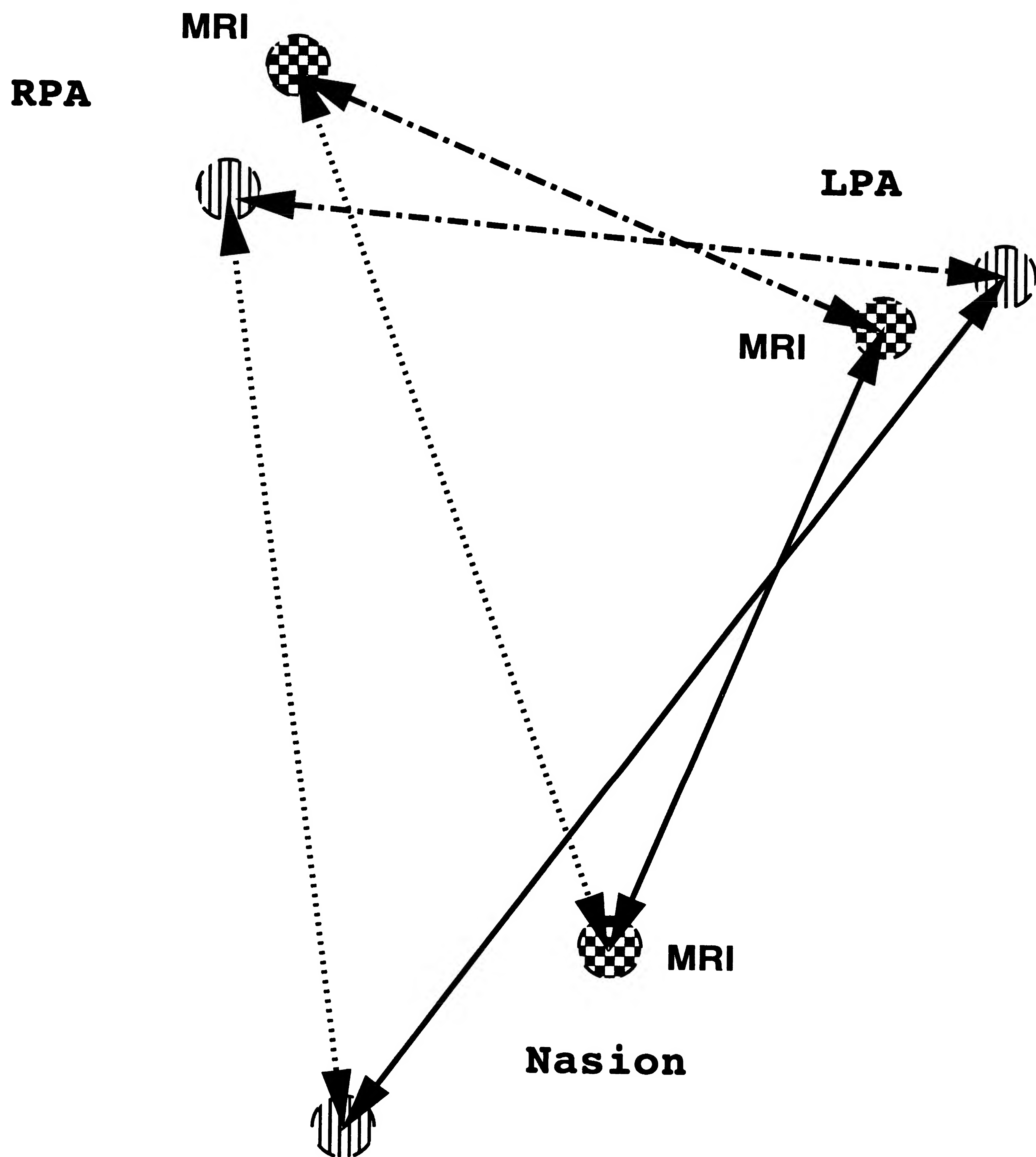


Figure 5.7: Warpage Percent Error is the difference in the lengths of the similarly styled lines.





If the measurements of cardinal landmarks in an MRI stack and the direct measurements with the Polhemus probe were exactly congruent, after transformation the locations of the two sets of cardinal landmarks would lie over each other precisely. Because there is error in the measurements, the triangles formed by the cardinal landmark triad are not the same shape. If one is only larger than the other, but have the same proportions, one triangle would have room to move around inside the other. This is an interpretation of “slop” as a type of backlash in the triangle congruency fit between measurement systems.

Unfortunately, current clinical MRI practice is to sample the study volume anisotropically. Within the slice plane, pixel resolutions are typically 0.8 to 0.9 mm. However, the distance between slice planes is typically 3 mm. When the data are assembled into a 3D array of pixel values or voxels, the distance between slices is not the same as the distance between pixels. The distance between the same pixel location between slices is not the same as the distance between pixels within a slice. The problem is one of relating array space to world space. Array space is the effective coordinate system the computer uses to access data in its memory. World space is the normal metric of reality; real measurements in real numbers. When the data are loaded into computer core (memory), each slice is represented by an 2 dimensional array indexed by an integer,  $i$  and  $j$ . Each slice in a stack of slices is an additional dimension to the main data array. Each slice is indexed by  $k$ . Any pixel can be referenced by the integers  $i, j, k$ . If our pixel array named `Pixel`, any location would be referenced as `Pixel[k][j][i]`. Nominally, the  $i$  direction refers to the horizontal or X direction, from the observers left to right, with X increasing from left to right. Each row  $i$  of the array `Pixel` is a contiguous array of pixel values. Similarly, columns of `Pixel` are referenced with the  $j$  index. Vertical shafts of pixels skewering down through a stack of slices are referenced with the  $k$  index. Stepping in any dimension one pixel is accomplished by incrementing or decrementing the  $k, j$ , or  $i$  integers. When we step by one count vertically, in the  $k$  direction within the stack of slices, we are moving 3 mm in world space. When we step in the slice plane one count, we are moving nominally 1 mm in world space. This makes the projection of the stack array space into world space stretched in the  $k$  direction with respect to the  $j$  and  $i$  directions. This implies that the error statistics are stretched in this projection also.

For analyzing errors in measurements taken from anisotropically sampled space, we must keep the anisotropic nature of our measurements in mind. Pedantically, precise disclosure of error statics should be given for each of the orthogonal dimensions. Our error statistics are actually best represented by an ellipsoid of uncertainty. The longest axis of our 3d ellipsoid of uncertainty would be in the  $k$  direction for the particular orientation of the  $1 \times 1 \times 3\text{mm}$  stack we are considering. What

this means is that for an axial stack, the  $i$  direction and the  $j$  direction correspond directly to X and Y. The  $k$  direction is the Z direction, and our ellipsoid of uncertainty is oriented with its long axis up, or in the Z direction. For a stack of coronal slices, the X direction is nominally in the slice  $k$  direction, and is the direction of greatest uncertainty. For a stack of sagittal slices, the slice  $k$  direction is oriented nominally along the Y direction. These are nominal directions because each stack can be reoriented in arbitrary directions to obtain coverage of anatomic features that may otherwise be outside the MRI field of view.

There are a number of drawbacks in the nature of the PPN System. It is not overdetermined and so provides no internal gauge of inaccuracy. In addition, is not possible to determine warpage error in the Z direction. Ideally, it should be possible to use as many landmarks as possible to spread localization errors in the measurement of each individual landmark point as evenly as possible. Adding a fourth reference point does not guarantee enhanced accuracy. For instance, a fourth pointer could be held against the top of the head during an MRI study, and its position in the stack could be compared with the position established directly with the Polhemus probe. However, preliminary theoretical evidence from Abche *et al.* [184] indicates that the reduction in error by using more and more registration markers becomes less and less asymptotic for more than 6 markers.

A principal advantage of the PPN system is that it is possible to have significant error in certain directions for each of the cardinal landmarks that have no effect on the orientation of the PPN axis. The nasion is insensitive to error in the X direction. The RPA and LPA cardinal landmarks are unaffected by errors in the Y direction.



## 5.5 The CoordAligner: A system to register stacks of image slices

The CoordAligner is a software tool to take stacks of data slices in one coordinate system and enable various measurements and manipulations so that 3D positions are specified in the PPN Headframe. An important feature of the CoordAligner is that it specifies the three classes of residual errors after registration of the cardinal landmarks, as explained in Section 5.4.2.1. This means that a bad fit can be recognized from the values of slop, warpage, and wobble, and data from a good fit can be used with confidence. The error distance between homologous landmark pairs after alignment is the worst case error from all of the landmark localizations.

One of the goals of the is software to enable users to make objective and repeatable measurements from stacks of slice image data. Different experienced observers of the same slice image should be able make the same measurements and obtain the same values. This has proven to be an exceptionally difficult goal to achieve. The main problem is that the various features we wish to measure, such as sets of marker traces, skin surfaces, and brain surfaces are indistinct and fuzzy. We have developed a number of interactive procedures allow an operator various options to semi-automate the interactive procedures to minimize hand tracing, pixel selecting, subpixel interpolation, and outlining as much as possible. These operations are highly subjective, and repeated measurements by the same or different operators yield different values. We can never eliminate operator variability, and an expert operator will always be able to obtain superior accuracy, and higher precision; but we can reduce the variability in measurements by providing as much assistance to the operator through interactive tools and facilities. Figure 5.11 shows the size of 1 millimeter pixels with respect to the Pointer registration marker traces in a typical marker trace digitization study.

The principal means of interaction with three dimensional volume data is through operations on a component slice in a stack of slices. The simplest interactions, and most straightforward interactions with the volume, are with the individual slices that make up a stack of volume data, specifically MRI slices. The PixelPicker is a subprogram in the CoordAligner system that displays 2 dimension slices and interprets mouse actions (button clicks and mouse movements) as 2 dimension locations on a slice and 3 dimensional space that the slice is embedded in. The 3 dimensional information such as feature locations on a slice, is used to assemble various points, lines, and locations in the space about a slice. However, obtaining a precise location

of isolated features from fuzzy image features having a size of a few pixels on a side is made easier with the Blobulator. Figure 5.9 shows a representative axial MRI slices and the segments of Pointer traces outside each of the ears.

Localization accuracy is enhanced by having the Blobulator program determine the position of maximum brightness for each marker spot or blob, thereby increasing the spatial resolution from the size of a pixel (3 mm ) to the subpixel uncertainty in determining the position of maximum weighted average brightness. The purpose of the Blobulator is to reduce the variability in determining the center of gravity of the blobby traces produced by the Vitamin E filled conical spoke shafts of the Pointers.

The Blobulator facility is a collection of methods to precisely and repeatably obtain positions from blobby or indistinct features such as Vitamin E capsule registration markers. The Blobulator calculates for first moment (a center of gravity) for a collection of bright pixels above a certain threshold. Figure 5.10 shows the contour outlines about a registration marker trace, and the associated center for that collection of irregularly shaped pixels.

Because locating points on slices depends on locating centers of blobby traces, we have developed a subsystem dubbed the "Blobulator". This uses a simple algorithm to find the center of gravity of a simply connected blob of pixels above some threshold. The operator seeds the Blobulator with a mouse click on some bright pixel on the blob, and the Blobulator marks each adjacent pixel above a threshold value until it reached the border of the blob. A blob boundary surrounds the interior pixels above the threshold, and the exterior pixels are below the threshold. A 2 dimensional contour plot delimits the blob border. Figure 5.10 illustrates the outline of a blob, and the weighted center of gravity of the pixels. This reduces variability in measurements from slices caused by different operators' hand, and makes measurements of blobby features more deterministic and repeatable. While the selection of the blob boundary contour threshold value is not precise, this value is extremely insensitive with respect to the change in the center of gravity position *versus* change in threshold value. This is clear to see for approximately symmetric blobby shapes at any contour threshold.



## *Pointer Traces*



Figure 5.9: A representative axial MRI slice, showing Pointers traces.



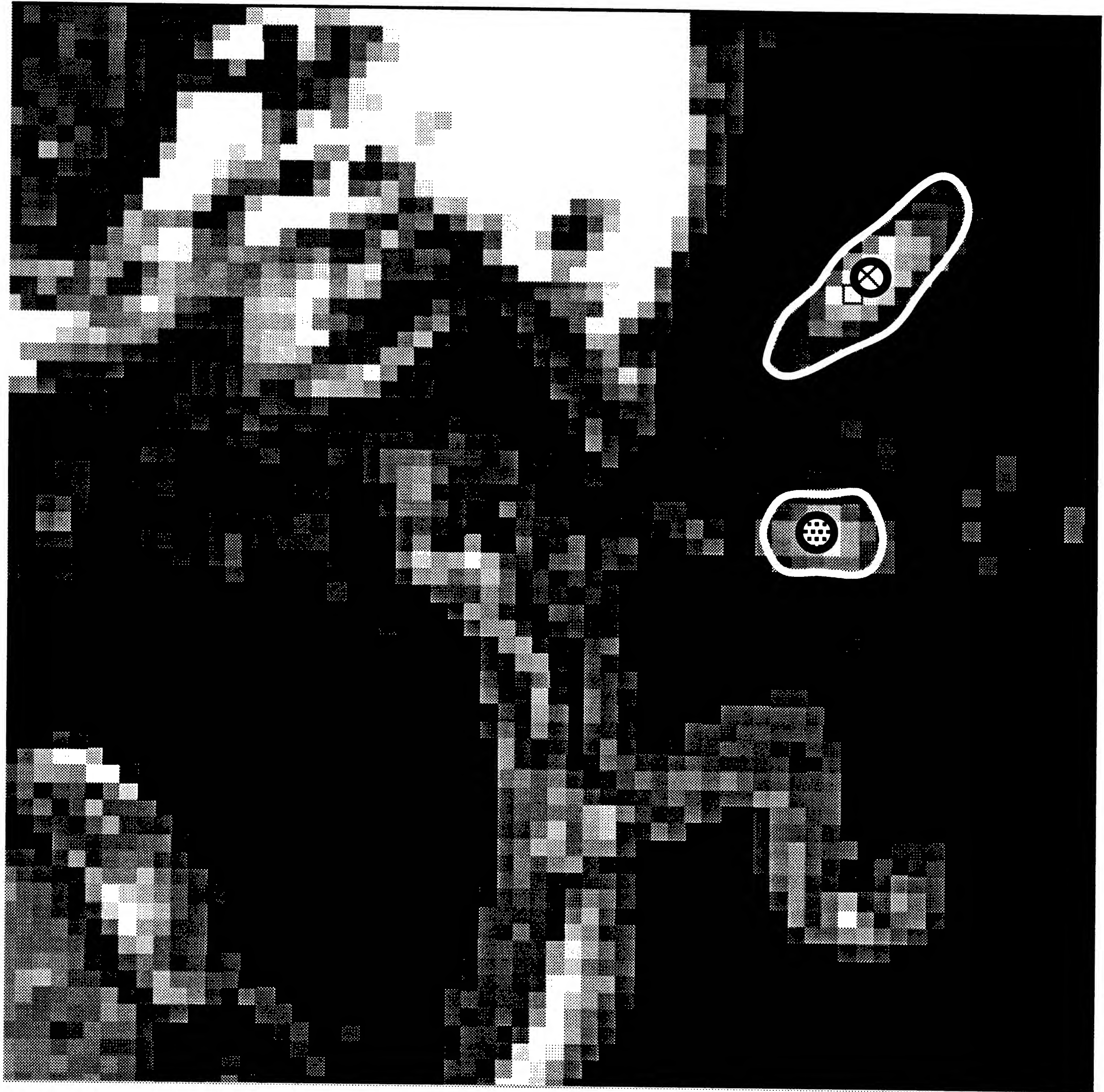


Figure 5.10: The Blobulator: magnified blob borders and the blob weighted center of gravity.



Two dimensional measurements on a slice, however precise and repeatable, will never be precise and accurate in three dimensional measurements. Two dimensional measurements are constrained to lie in the slice plane, while the object being measured is not. The goal of sighting a registration marker to locate the marker in three space, and register the object. The locations measurable on any slices will suffer from aliasing by the slices. All of the measurements will necessarily cluster about the slice planes, and cause errors (aliasing) and artifacts in registering the associated object. Another problem with this planar measurement approach is that the landmarks may be hard to see in the orientation of a particular stack of slices. Figure 5.11 show the appearance of nasion in a sagittal slice. This is highly visible landmark in the stack. However, for the left and right periauricular points, approximating their position from slices is more difficult. An experienced observer can make good ( $< 3$  mm), localizations of the cardinal landmarks by estimations of blob centers, as Table 6.7 shows. The probability of a slice passing through the desired landmarks and its associated markers, increases with the number and density of slices. We show in the next section a simple technique to interpolate between slices in a stack of MRI images with the Pointers technique.

### 5.5.1 Pointers for Localization of Cardinal Landmarks

Pointers are devices that leave traces in a stack of MRI slices that enable precise interpolation of the cardinal landmark locations that may not be visible from an individual slice in a stack of MRI slices. When a MRI image is recorded, three Pointers are held against the skin to “point” to the cardinal positions. The Pointers are worn as shown in Figure 5.4. As shown in Figures 5.2 and 5.3, each Pointer consists of a Plexiglas block in which 4 hollow tubes have been blind drilled so their axes converge, analogous to spokes of a wheel with a conical geometry, to a point that is precisely centered over the place the Pointer rests on the skin. The tubular spokes stop a millimeter short of the point where their axes intersect, so that each end is blocked. They are filled with a material that can be imaged by the modality of interest, and a cap is secured to the outer end. For MRI recordings, the spokes are filled with Vitamin E, so they may be imaged by recording additional slices outside the head. For CT studies, chalk, lead, or bone meal could serve as a contrast agent for X-rays. For PET and SPECT studies, radioactive  $^{68}\text{Ga}$  in the Pointer’s spokes would be visible by their emissions.

An adjustable arm that can be made rigid permits convenient placement of the pointer over a cardinal landmark. This is illustrated in Figure 5.4. All three

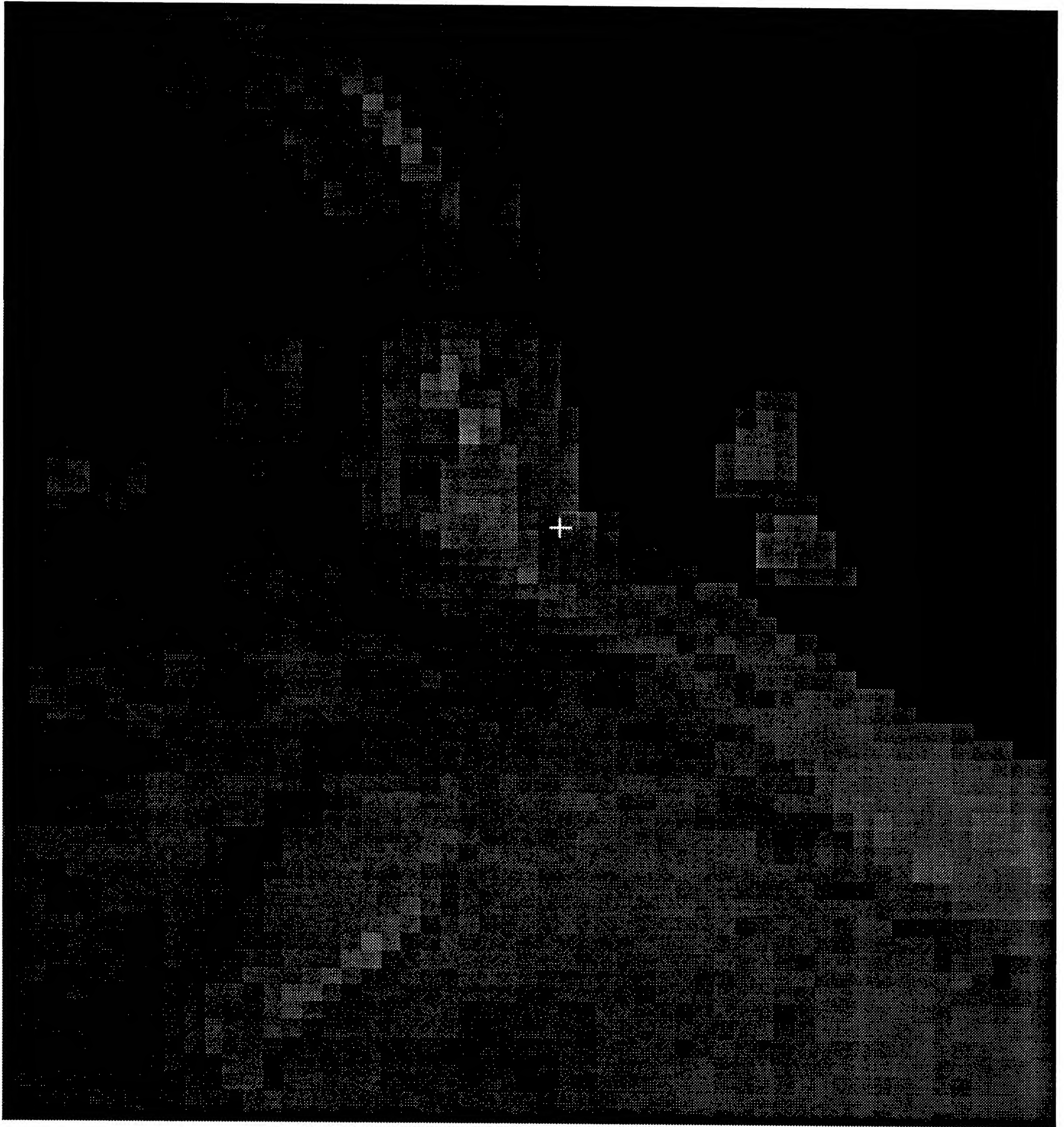


Figure 5.11: The appearance of nasion and the blobs of two spokes of a Pointer, in an sagittal MRI slice.



Pointers can be attached to a headband made with Velcro<sup>TM</sup> or to a plastic welder's head band which is modified to support clamps that hold the rod for each Pointer. It is important that the Pointers be kept as small as possible, since imaging devices typically do not provide much additional space around the subject's head. For instance, when recording MRI scans the head coils are only a few centimeters from the scalp, to enhance sensitivity. The Pointers used in our laboratory extend 5 cm from the scalp, which provides only a few millimeters of space for the subject head to be positioned in the MRI bore.

The 4 spokes of each pointer shaft are oriented so as to permit them to be registered as a sequence of spots, for any orientation of the stack of slices. Thus, the same set of Pointers serves for axial, coronal, and sagittal slices. The corresponding images appear as a sequence of elliptical blobs, as portrayed on a the PixelPicker, in Figures 5.14, 5.15, and 5.16, the pattern of spots converges toward the position where the Pointer rests. The user interacts with the software and slice image to construct a system of lines through the blobs corresponding to each spoke of the Pointer. The software calculates the point in space, between the slices, that is closest to the system of the lines. The Blobulator finds the center of moment point in a blob, the BestLine algorithm subroutine finds the best line through these points, and the BestPoint algorithm subroutine finds the best that best solves the system of lines. We assume that the points identified in slices do not line up precisely along consistent lines, and that the lines do not converge at a point. Therefore, we find the best line, closest to each constituent point. Similarly, we find the best point, closest to each constituent line, to interpolate between slices. The mechanics of these geometric computations is given in the BestLine and Point algorithm section, below.

Once the positions of the three cardinal landmarks are established as described above, the next step is to construct the PPN Headframe from the landmarks. This is established with respect to the MRI system coordinate system by the definition given above. A transformation matrix is constructed so that any  $i, j, k$  integer pixel or subpixel MRI location on a slice can be expressed by the value of  $X, Y, Z$  cartesian coordinates in the PPN Headframe.

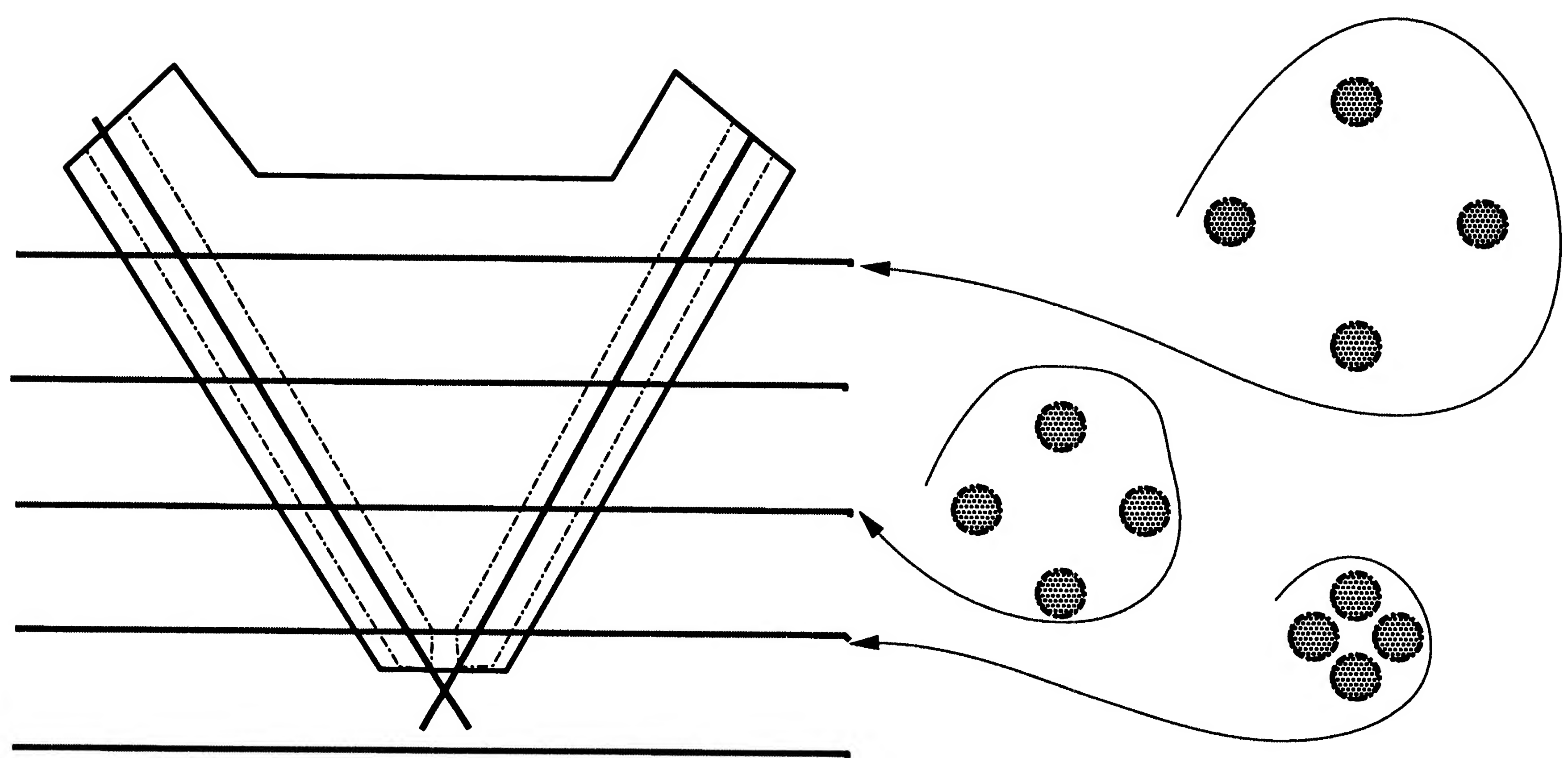


Figure 5.12: The intersecting planes of the MRI slices and the Pointer contrast media chambers. The resultant pattern of dots as it appears in the MRI slice plane is shown for each intersection.

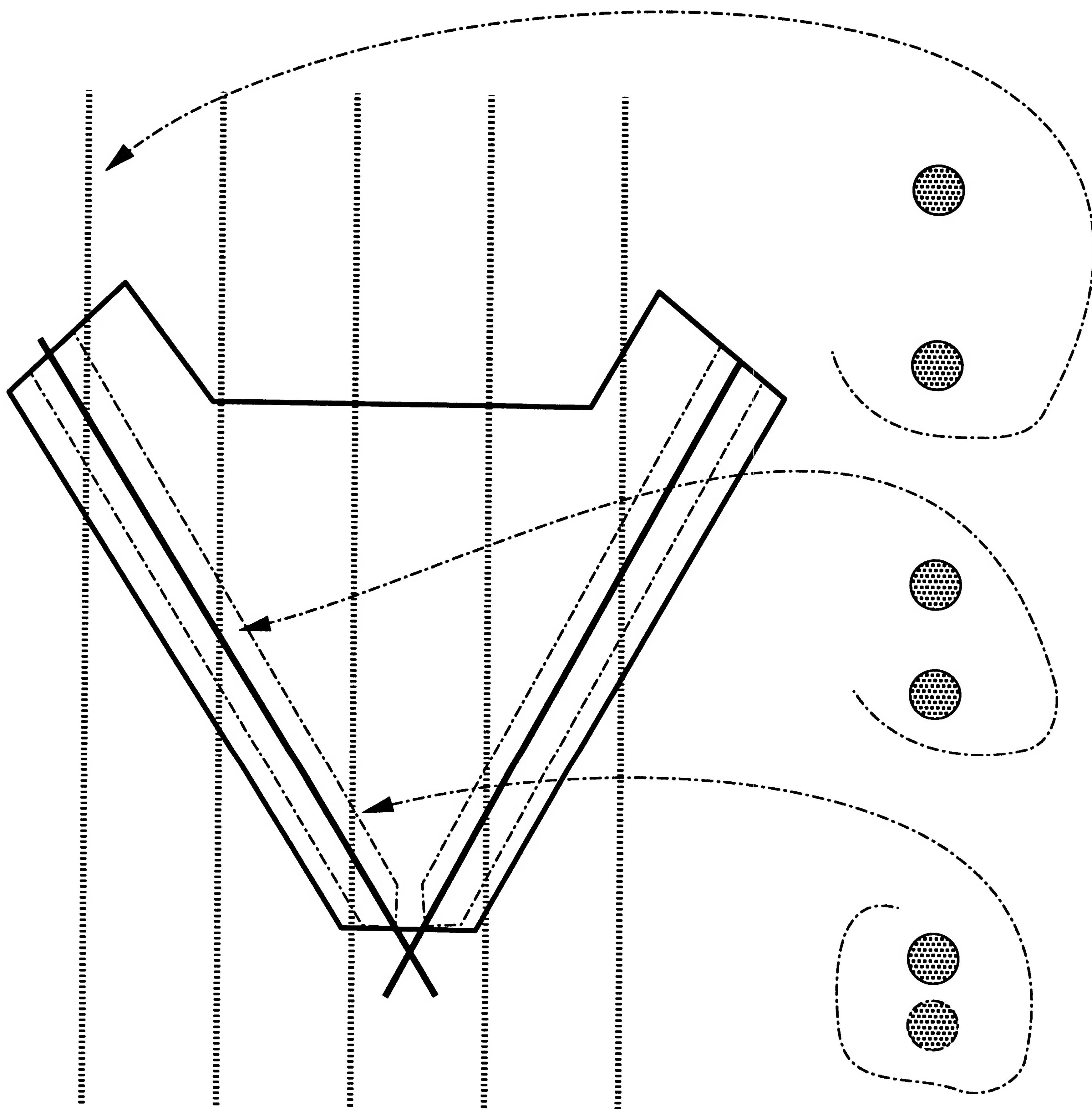


Figure 5.13: The Pointers intersection from vertical slices, with the resultant pattern of dots.



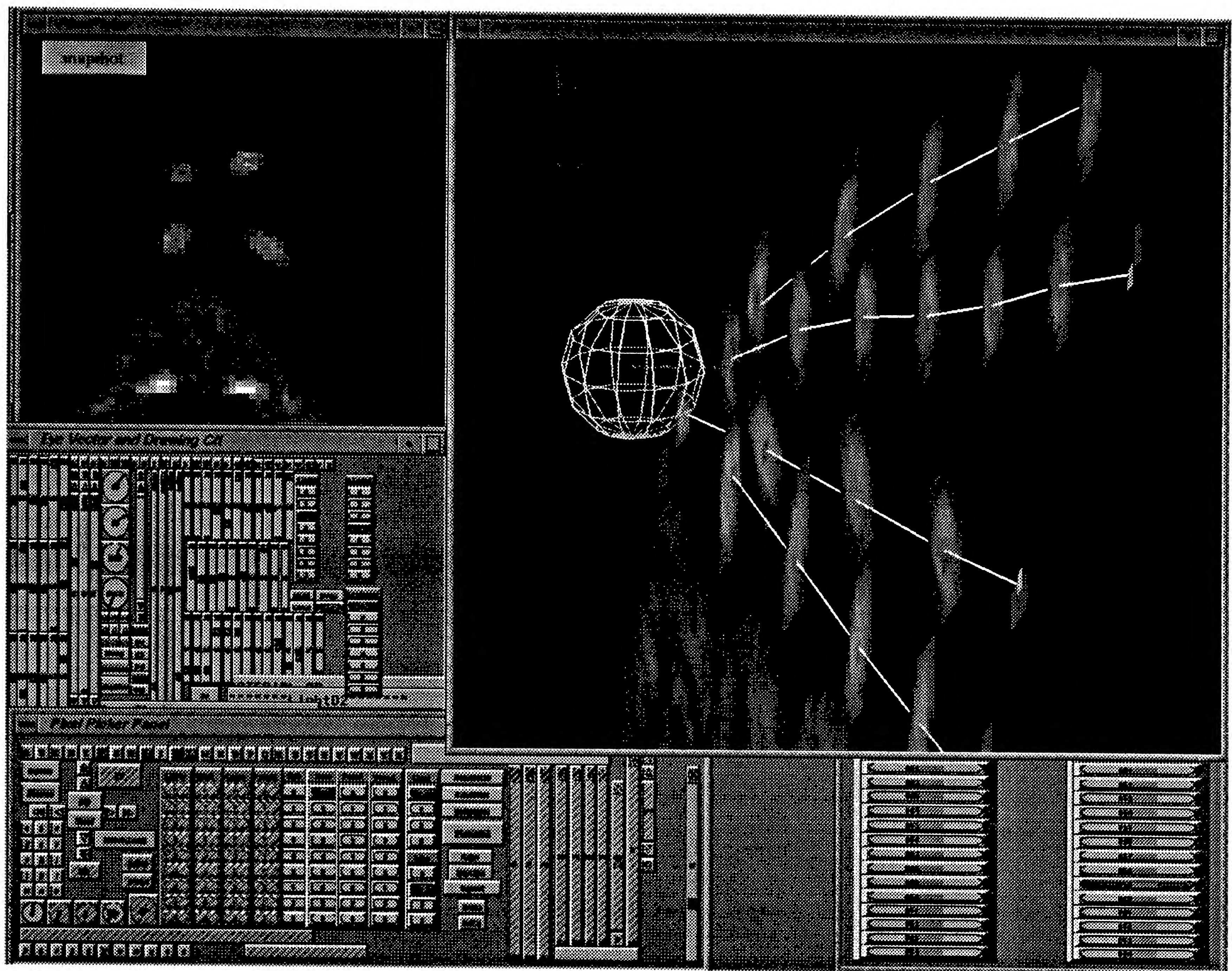


Figure 5.14: CoordAligner software panel showing line fitting operation through the blobby pointer traces.



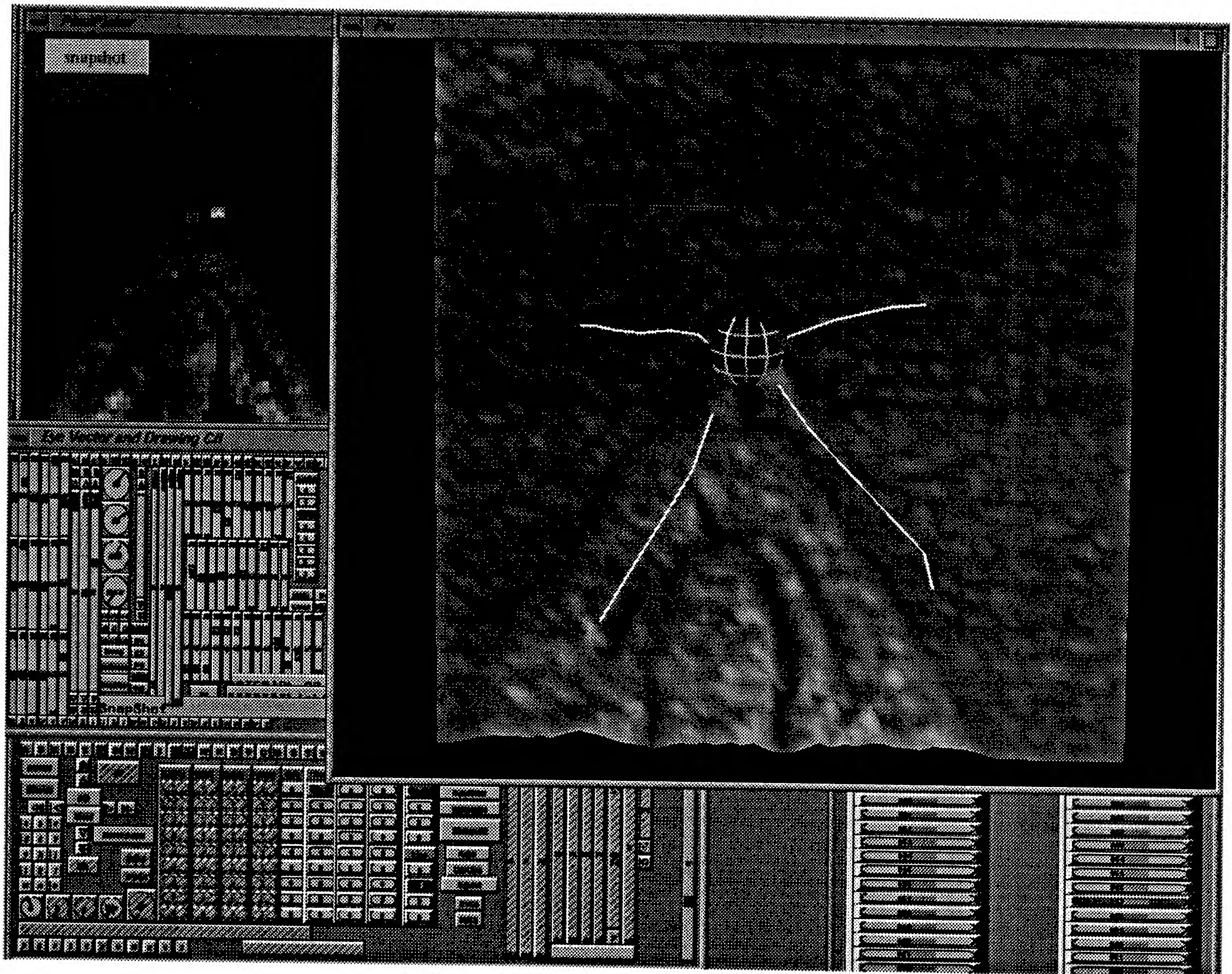


Figure 5.15: CoordAligner locating nasion from best lines through nasion Pointers contrast media chambers.



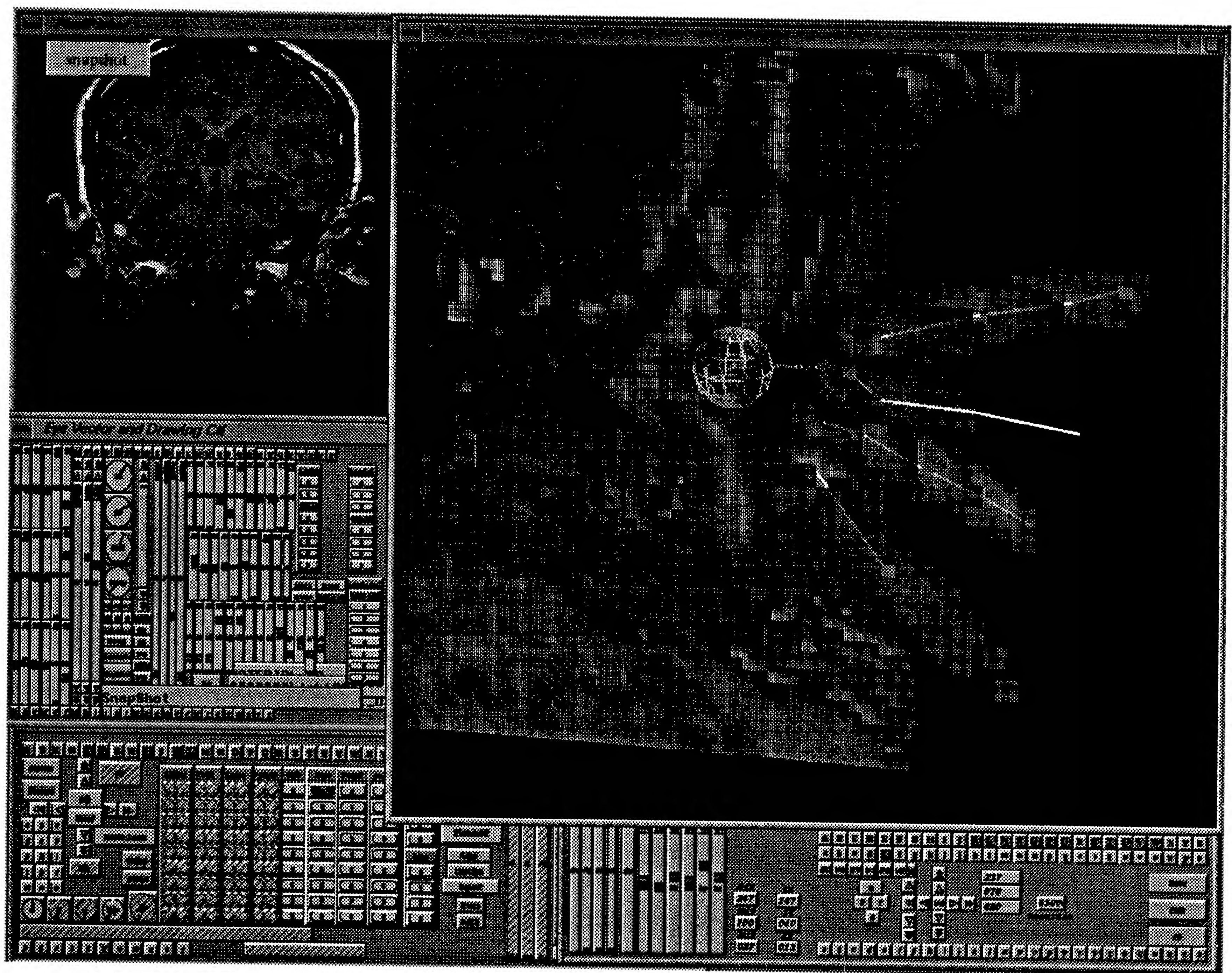


Figure 5.16: CoordAligner locating cardinal landmark from Pointers over right periauricular point.



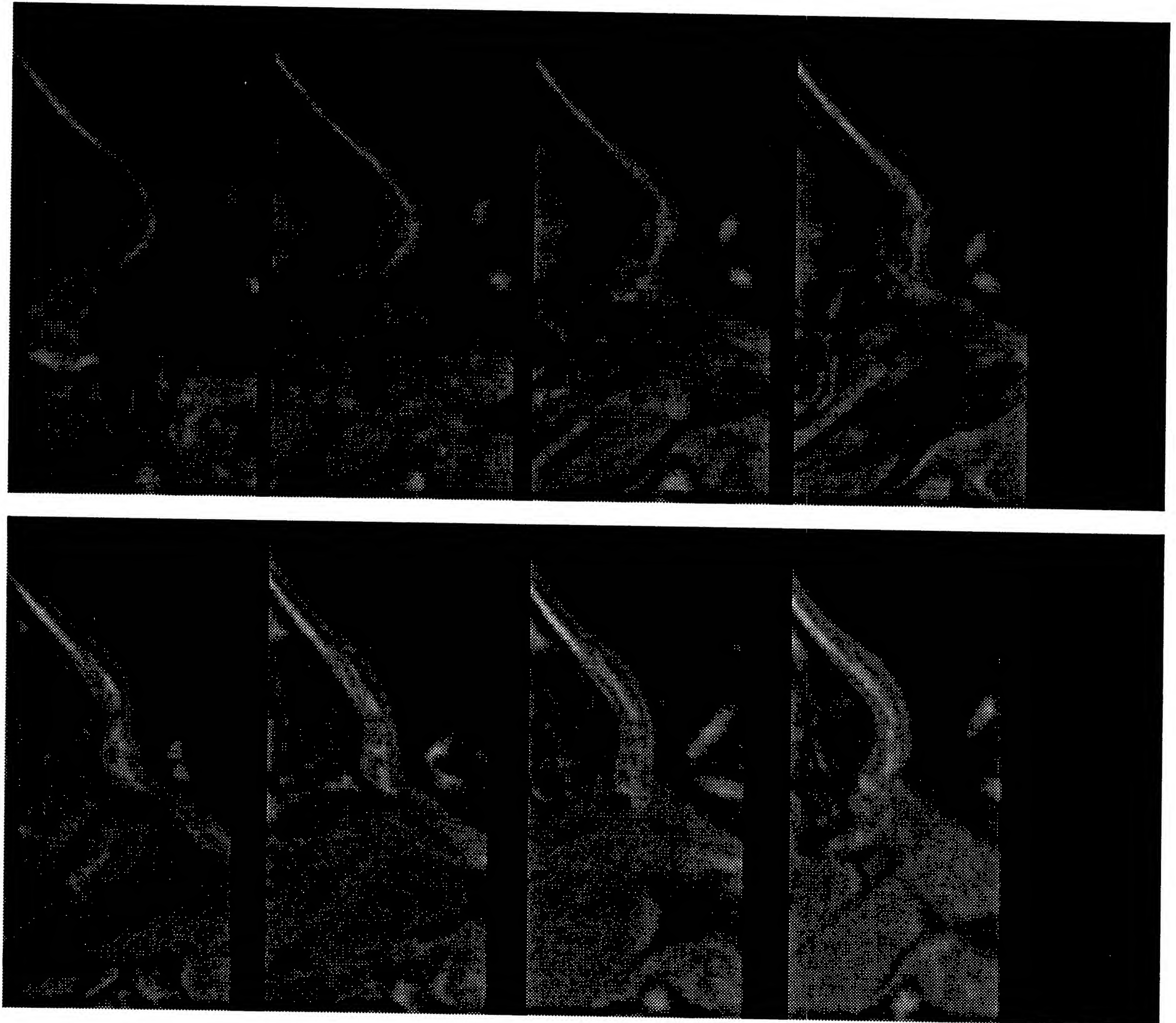


Figure 5.17: Pointer traces in a sequence of sagittal a slices through nasion.

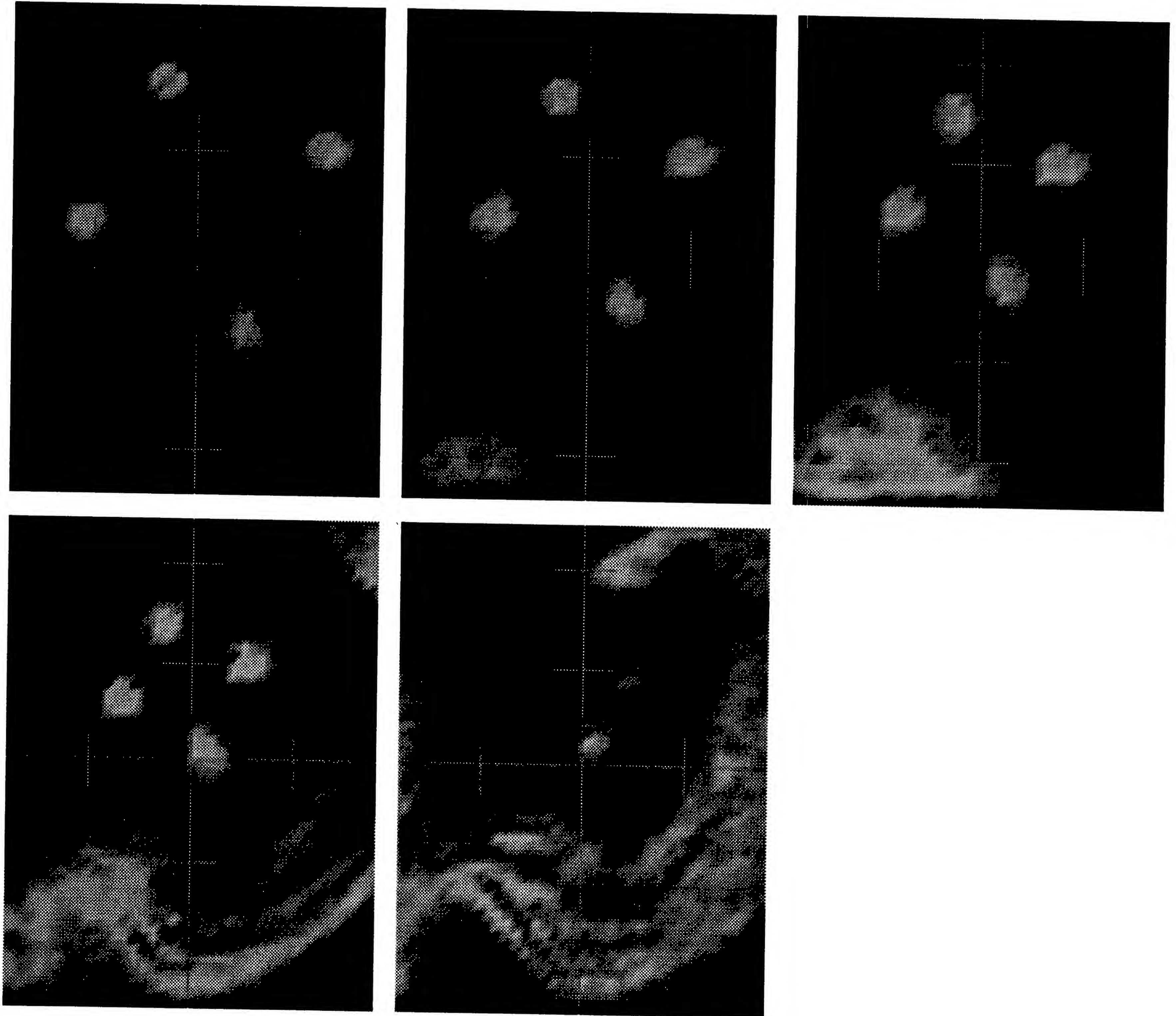


Figure 5.18: Pointer traces in a sequence of sagittal a slices over a periauricular point. Distance between axis ticks are 1 cm.



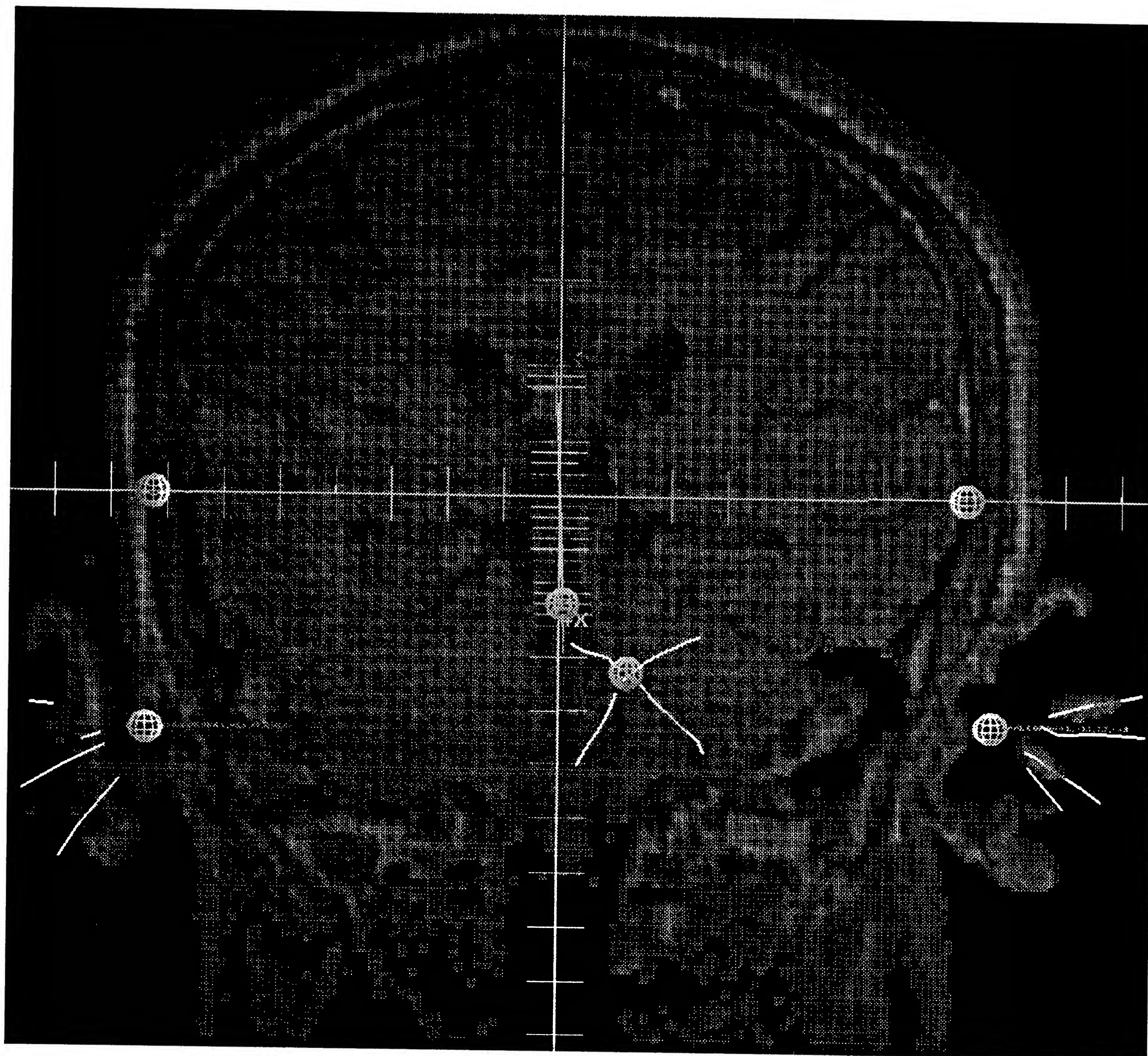


Figure 5.19: A coordinate system before alignment to cardinal landmarks.



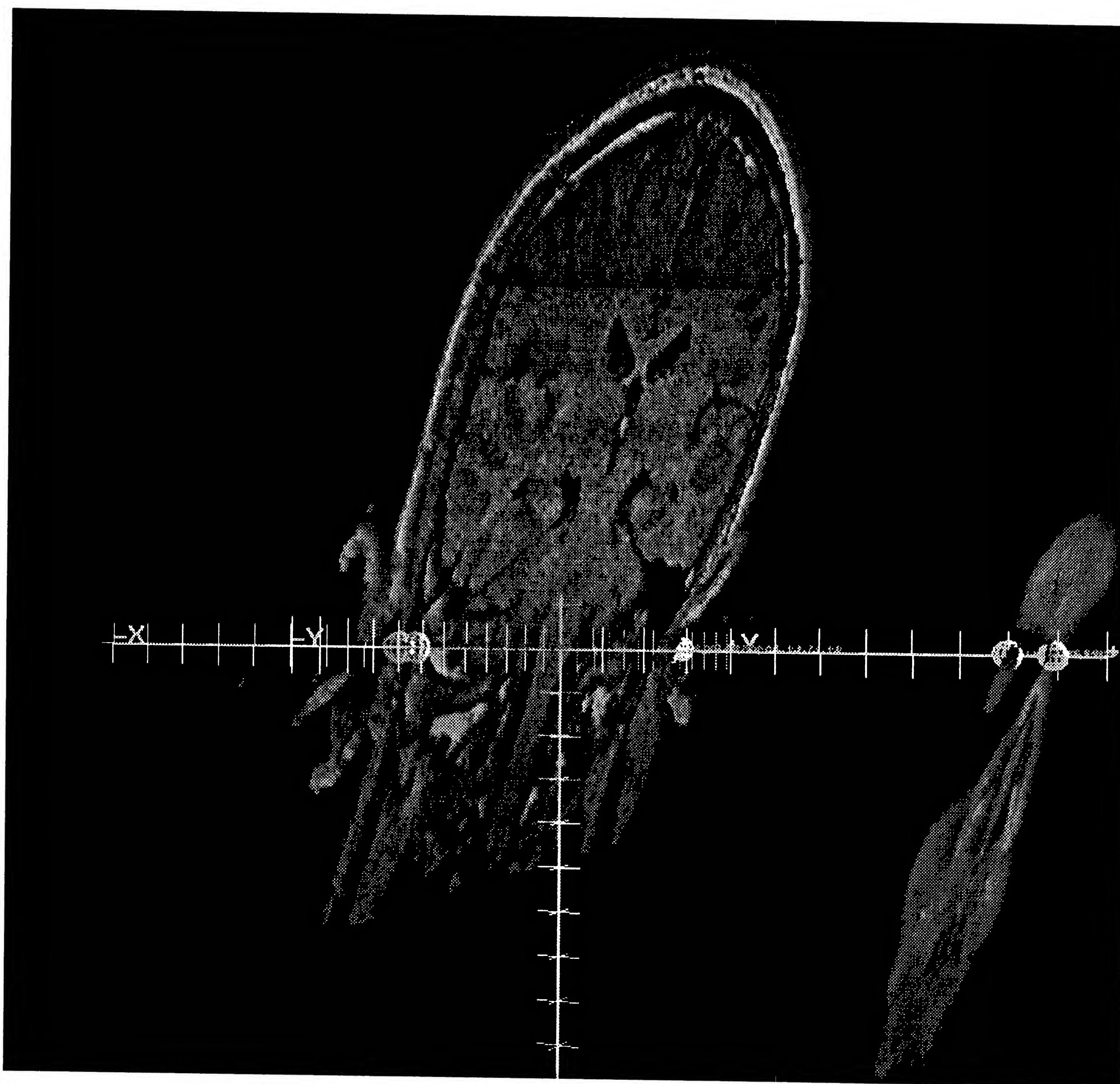


Figure 5.20: After registration, the slices are tipped into the head-frame coordinate system. Two slices are shown, the right slice passing through the nasion

### 5.5.2 Pointers Algorithm

The Pointers provide image traces that can be assembled into lines. From this system of lines a point calculated that best solves the system of radial spoke lines. The calculation of the best line through a system of blobby trace centers, and the calculation of the point passing closest to a system of lines is presented in this section.

The Pointers best line and point algorithm can be stated mathematically as the solution to these two problems. Given a list of points, find the system of equations for the best line that passes a minimum distance from each point. Given that list of lines, find the point that is a minimum distance from each line. Singular value decomposition (SVD) is used to solve for the inconsistent matrix. The system of equations is almost always inconsistent because noise in the image and unavoidable digitization errors will produce a list of points that do not lie on a straight line, and a list of lines that do not pass through a common point. The system of lines are said to be skew if they miss touching each other. There are three alternate representations for a line in three dimensional space. Each equivalent form is tested for numerical robustness and the best solution form is used. This avoids singularities discovered in the best fit lines that are oriented along certain canonical directions, or pointed in directions close to the axes. This algorithm is implemented in the "C" language. The Pointers algorithm been used in locating points in tomographic slice stacks, and finding trajectories of high energy particles passing through arrays of detectors. The source code is presented in the Appendix, Section A.

The two problems we are going to solve are :

- 1 Find the line that has the minimum total distance from a list of points denoted  $(x_i, y_i, z_i), i = 1, 2, 3, \dots$
- 2 Find that point that has the minimum distance from a given list of lines.

It is helpful to represent each line as the intersection of two planes. The solution of the second problem then follows simply from the results of the first problem. Each line is given as a pair of equations that specify the intersecting plane equations:

$$(5.16) \quad A x_i + B y_i + C z_i = 1$$

$$(5.17) \quad D x_i + E y_i + F z_i = 1$$

where the entire set of planes is specified by  $i = 1, 2, 3, \dots, n$ .

The first problem can be formulated as

$$(5.18) \quad [\mathbf{A}] [\mathbf{X}] = [\mathbf{B}]$$

where

$$(5.19) \quad [\mathbf{A}] = \begin{bmatrix} x_0 & y_0 & z_0 \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix}$$

$$(5.20) \quad [\mathbf{X}] = \begin{bmatrix} A & D \\ B & E \\ C & F \end{bmatrix}$$

$$(5.21) \quad [\mathbf{B}] = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix}$$

Note that the data matrix  $[\mathbf{A}]$  has, in general, many more rows than columns, so that this system is usually over determined. An over determined problem may have no solutions. The Singular Value Decomposition (SVD) solution is the best possible solution and can has the minimum RMS error properties.

The Singular Value Decomposition (SVD) technique is based on the observation that any matrix  $[\mathbf{A}]$  whose number of rows  $N$  is greater or equal to its number of columns  $M$ , can be written as the product of an  $N \times M$  column orthogonal matrix  $[\mathbf{U}]$ , an  $M \times M$  diagonal matrix  $[\mathbf{W}]$ , with positive or zero elements, and the transpose of an  $M \times M$  orthogonal matrix  $[\mathbf{V}]$  [204], [238], [242]:

$$(5.22) \quad [\mathbf{A}] = [\mathbf{U}][\mathbf{W}][\mathbf{V}]^T.$$



The “standard” SVD solution, is given as:

$$(5.23) \quad [\mathbf{X}] = [\mathbf{V}][\mathbf{W}]^{-1}[\mathbf{U}]^T[\mathbf{B}]$$

SVD solutions are similar to the Moore–Penrose pseudo-inverse and are essentially the same as a generalized constrained minimum or least squares solution [242]. The main feature of SVD is its stability and robustness with ill conditioned matrices. In our case, ill conditioned matrices are produced by lines described by pairs of planes that are almost parallel. This leads to severe numerical instabilities if these values are used to solve the second problem. To minimize these numerical instabilities we constrain the solution planes to be perpendicular. It has been my experience that any calculation with these lines will suffer from propagated numerical instability if the rows of the line matrix are not “maximally” linearly independent.

Intuitively, think of the line as two intersecting planes. Think of each plane as minimally defined as three points. Now at first glance it would appear that 6 points are required to define two planes. Remember that the planes can have points in common, and that we can take two triangles for two planes that share two points. So we can minimally solve for two planes in space with 4 points. If we now constrain our planes to be perpendicular, then we do not need one point. If we also allow our perpendicular planes to rotate about the line, then we do not need the remaining third point to constrain the perpendicular planes in any orientation with respect to our line. Therefore, we can use two points to define a line, where the orientations of the planes are undefined but perpendicular to each other.

From the above argument, it is clear that there are multiple equivalent pairs of planar equations for any line. Maximal linear independence implies perpendicular plane equations. Specifically, perpendicularity for a system of  $n$  planar equations means that any one coefficient from the first plane in Equation 1,  $\{A, B, C\}$  and any one coefficient not in the same column  $\{D, E, F\}$  can be 0. This is equivalent to projecting the line onto any two of the  $x-y$ ,  $y-z$ , or  $z-x$  planes. This reduces the  $2 \times n$  overdetermined equations in 6 unknowns to a three systems of  $n$  overdetermined equations in two unknowns. However, if any one of the  $A_i, B_i$ , or  $C_i$  values or the  $D_i, E_i$ , or  $F_i$  for our planes approach zero, then one pair of equations becomes singular. To avoid singularity in any particular direction, we must determine which pair from the three pairs of possible solution equations to use. Recall that the equations are homogenous and normalized, and since each equation solution must be equal to 1, any difference from 1 is symptomatic of approaching singularity. This difference is effectively a simple condition index for the matrix. We throw out the equation pair with largest error, and keep the two pairs of equations with smaller error. A line that

has a direction that approaches singularity will have a bigger numerical error in a trial solution than the others. However, we do not know *a priori* which of the systems of equations to use. The key idea in this new algorithm is to try all three systems and throw out the solution system with the largest error. The algorithm does not throw out any lines, only the equivalent representations of the line with the largest error.

The solution of the second problem of finding the point that is closest to a collection of skew lines is given simply by

$$(5.24) \quad [\mathbf{A}]\hat{\mathbf{X}} = \hat{\mathbf{B}}$$

Where

$$(5.25) \quad [\mathbf{A}] = \begin{bmatrix} A_0 & B_0 & C_0 \\ D_0 & E_0 & F_0 \\ A_1 & B_1 & C_1 \\ D_1 & E_1 & F_1 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ A_n & B_n & C_n \\ D_n & E_n & F_n \end{bmatrix}$$

$$(5.26) \quad [\mathbf{B}] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix}$$

$$(5.27) \quad \hat{\mathbf{X}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$\hat{\mathbf{X}}$  denotes the single point that best solves  $[\mathbf{A}]$ , *i.e.* the point minimizing the distance to all of the lines. The point can be interpreted as the singular value decomposition solution of the best solution to the inconsistent system of planes describing the lines.

We evaluate the error in each line and the point calculated with this algorithm by simple back substitution of the solution into the problem. The sum of the distances from each constituent point back to the best line gives us an index of the quality of the line fit. The algorithm will distribute error as evenly between each point, as each line is the unweighted solution to that line. The accumulated distance from the best point to each skew line gives a figure of merit for how skew the lines are. These figures are calculated between each mouse click, and displayed by the CoordAligner. This provides important feedback as to how well or poorly each point contributes to a precise or imprecise localization of the landmark at the apex of the Pointer spokes. The Appendix, Section A.4, contains a sample report showing the various errors of a pointer registration run. Experiments to determine nominal accuracy and precision of the overall technique are presented in Section 6.

The power of the Pointer technique is its extreme robustness for noisy images. The geometry of the Pointers is precisely defined, and computer models in Figures 5.3 and 5.2 show the idealized shape of the cylindrical spokes that contain the contrast media. A three dimensional SpiderWeb surface reconstruction of the Pointers is shown in Figure 6.10. It shows the isodensity surface of the pointer traces obtained from a frontal MRI stack. The nose extends down the left side of the image, and the eyes and eye-lids extend to the right. The Pointers are visible as four conical spokes emerging from the bridge of the subject nose over the nasion cardinal landmark. The pixels making a dark sawtooth emergence from the skin just to the right of the Pointers designate the reference slice about which this surface was constructed. The blobby traces we see in Figure 5.10 do not render into a clean cylinder. There is some small scale geometric distortion that gives the pointer spokes a wobbly appearance. This effect can be eliminated by smoothing the surface, if desired. This is due to the small signal strength from the small contrast medium volume within the borders of a MRI slice. The roughness of the rendering is not important to the mathematical pointer analysis just described. The Pointers construct a point from a much larger volume. Random errors in individual trace brightness, border, and position will tend to average out in the line and point construction. A single point is constructed from



generally 4 lines, each piercing as many as 5 slices, to yield 20 blob traces. This gives the Pointers technique its robustness.

An additional problem with the Pointers is that they can move off the cardinal landmarks if not held securely. A secure head frame for positioning the Pointers is made from a welders helmet plastic webbing. Extreme care and attention is required by the MRI operator not to move them positioning a MRI head coil over the subject's head.

The reduced clearances about the head caused by the new generation of MRI machines with their helmet-like close fitting head coils make use of the Pointers problematic. The new techniques in recording the echoes produced by radio frequency RF depolarization of protons require a pickup coil to be placed as close to the re-phasing echo source as possible [167]. This means putting a antenna almost directly on the scalp. The new Siemens MRI systems at New York University make extensive use of special head coils for high resolution, high sensitivity MRI studies. The next section explores the use of a geometric model of the skin surface to locate cardinal landmarks using the SpiderWeb algorithm to build a skin isosurface.

## 5.6 Visual Localization of Cardinal Landmarks Provided by SpiderWeb

The purpose of using a computer to determine cardinal landmarks is to match locations sighted on the skin surface by the neuromagnetometer operator with locations registered in a stack of MRI slices. It would seem reasonable that if we could recover the geometry of the skin surface from the MRI slices, we could use visual clues provided by computer graphics to find cardinal landmarks with the same heuristics as used by the neuromagnetometer operator. This would eliminate the need to establish fiducial markers with external hardware. This next section outlines the use of the SpiderWeb algorithm to construct a surface through a stack of slices in an arbitrary orientation. The skin isosurface can be constructed, and the cardinal landmarks located by inspection on a surface rendering of sufficient quality. Accuracy is enhanced by comparison with reference photographs of the cardinal landmarks as sighted by the neuromagnetometer operator. A special surface cursor, called a bug, is constrained to lie on the constructed surface. The bug is constrained to move about on the surface. This bug can be used to probe features on the surface geometry, and associate various surface locations with cardinal landmarks.

A major feature of the SpiderWeb algorithm is that it is fully functional with less than perfect MRI scans. Generally, clinical MRI studies are of this nature. Figure 1.6 shows a representative high resolution MRI scan with noisy artifacts present. The artifactual surface about the eye, resembling a mask, are caused by eye ball movements, probably REM sleep eye movements. The halo about the top of the head is an analytic continuation artifact of the MRI 3D inverse fourier transform. It causes features at the bottom of a stack to repeat them selves at the top of the stack. It is analogous to a weak horizontal hold on a TV reception, causing the bottom of the picture to repeat at the top of the screen. The main point of this study is that this surface is till navigable by a local surface probe, or bug cursor.

We can consider the surface to have a local two dimensional coordinate system,  $U, V$ , which is the surface space. The “bug” is a spatial cursor that is constrained to move on the SpiderWeb surface. The bug can move in the  $X, Y, Z$  coordinates of the three dimensional MRI space. However, it is not free to move about in any direction in the  $X, Y, Z$  space if it is to remain on the surface. Manual steering in  $U, V$  space is difficult because we have found that despite being constrained to move on the surface, the operator intuitively thinks of moving the bug in the three dimensional MRI space. However, in world space the bug can become stuck in a *cul-de-sac*. An



example of this would be trying to move the bug upward (in the  $+Z$ ) direction when stuck at the top of the head or the top of the ear pinna. There is no place to go further up, and it remains stuck at an apex of skin surface. Because the SpiderWeb isosurface has well defined topological properties [140] [141], such as the surface is manifold, the bug can be steered by hand and with guidance from automatic heuristics for homing to a cardinal landmark.

The robot spider is also capable of being programmed. Automatic navigation of the robot spider has been achieved in a limited way. The bug needs to be started by the user at some location on the surface. It can be programmed to scan the local surface, make a calculation of some local property for each of the adjacent sections of surface it can step into, and move in the direction with the greatest merit property. The merit property is some function that returns a value for each choice of adjacent triangle that the bug has available to step into. A simple merit function is to move in the  $+Z$  direction. It examines each adjacent triangle and the merit function returns the  $Z$  value of the centroid of the triangle. The triangle with the largest  $Z$  value is selected, and the bug moves into that triangle. Other example merit functions are direction vectors that the bug tries to move as close to as it can, constrained by the surface it is on. An average curvature merit function returns a region's curvature for each adjacent triangle, and the bug will tend to move in the direction of greatest curvature. Special heuristics for bubble bifurcations keep the bug from getting wedged inside a bubble, or buried in foam. Essentially, it will not step into a triangle whose normal vector has a negative scalar product with respect to the current triangle the bug is on.

However, as can be seen by the surface renderings in Figure 1.6, the surface can have considerable noise and roughness. This can cause the bug to become wedged into a local minima. The best way out of this problem, for the scope of this study, is for the operator to steer the bug out of a *cul-de-sac* and go to an appropriate location. The successful navigation over a terrain by purely local information is an interesting area of robotics research. However, we shall localize the bug with the operator's global viewpoint until a suitable algorithm becomes known.

The skin in MRI generally has a strong density gradient. MRI signals represent microvolts of radio-frequency echo signal from each pixel. The pixels outside the skin, in the surrounding air have a small, weak signals, and thus appear dark. The strong signal from the protons in the skin appear as bright pixels. What this means is that the isosurface of the skin has very low values on the immediate outside, and very high values on the inside, on the order of 50% to 80% of the maximum histogram value for a region of interest. We have seen that the selection of the



particular isovalue for skin is not critical for the position of the skin, as long as it produces a clean surface. Additionally, we have shown that the PPN Headframe is insensitive to position errors perpendicular to the skin surface. Since changing the isovalue over a small range of values would only contract or expand the skin surface relative to some coordinate system origin, normal to the axis, we can see from purely mechanical grounds that this would not affect the coordinate system position or orientation. Changing the isosurface value 10 or 20 percent moves the isosurface only 20 percent of the distance of a voxel edge. For a  $1 \times 1 \times 3$  mm stack, the distances are proportionate to the gradient or difference. For a strong difference, a small change in isosurface value makes a smaller difference in position than a weak gradient. The estimation of the “best” skin isosurface value can be calibrated by fitting a skin isosurface to the known cardinal landmark distance triangles, but this assumes no MRI warpage. Further studies in this area are in order, but beyond the scope of this project.

## Chapter 6

# Results: Accuracy of the Pointer and SpiderWeb Registration Method

A comparative study of three localization techniques for cardinal landmark was carried out with three approximately orthogonal stacks of MRI slices for each of two subjects. The three techniques were: the Direct Method, where cardinal landmark were visually selected directly from a slice, and only on the plane of a particular slice; the Pointer Method, where the spoke traces from each pointer were located on slices, and the precise position of the cardinal landmark at the intersection of the spokes was calculated; the “Spider In Web Method”, where a steerable robot “Spider” cursor was interactively navigated about a small region of skin isosurface until it appeared at a location that was judged to correspond to the skin location touched by the Polhemus digitizer probe. Each study consisted of using acceptable MRI stacks obtained from the Philips Gyroscan 1.5 T system at the New York University Medical Center. An acceptable study is a low noise, high contrast spin echo MRI study that included nasion, left periauricular and right periauricular point in each stack of slices, and at least two acceptable orthogonal stacks. Each acceptable stack also had to enclose a functionally significant region of brain where neuromagnetic data would be registered. Typically, this included auditory cortex or visual cortex.

Of a total of 5 persons studied (HW, LK, DK, ZL, SW), only two studies were sufficiently complete to permit a registration study. An incomplete study was defined as a MRI study that lacked a complete set of cardinal landmarks, or the

Pointers identifying them, in the stack of slices. Incomplete studies were rejected. The field of view of the MRI system is the 3D volume that is imaged in a stack of MRI slices. This is typically a cube of approximately 20 cm on a side. An example of an incomplete study is where the nasion was visible in the coronal stack, but the left and right periauricular points were not. A composite of stacks can be registered because the orientation of each stack was known with respect to the MRI bore coordinates. The CoordAligner software includes algorithms to register information that may be distributed in multiple stacks. In this way if the nasion is visible in a coronal stack, and the left and right periauricular points are visible in a sagittal stack, the two stacks could be registered with respect to each other and a complete set of landmarks composited together. The individual stacks were rendered in three dimensions within the MRI coordinate system, and the locations of each cardinal landmark from each stack was used to calculate an alias transformation matrix for the entire composite. However, as the skill of the MRI operator increased, we were able to obtain a complete triad of landmarks in each stack with only a fine adjustment of the MRI field of view orientation.

The field of view of the MRI system is the 3d volume that is imaged in a stack of MRI slices. This is typically a cube of approximately 20 cm on a side. An example of an incomplete study is where the nasion was visible in the coronal stack, but the left and right periauricular points were not. A composite of stacks can be registered because the orientation of each stack was known with respect to the MRI bore coordinates. The CoordAligner software includes algorithms to register information that may be distributed in multiple stacks. In this way if nasion is visible in a coronal stack, and the left and right periauricular points are visible in a sagittal stack, the two stacks could be registered with respect to each other and a complete set of landmarks composited together. The individual stacks were rendered into MRI coordinate system, and the locations of each cardinal landmark from each stack was used to calculate an alias transformation matrix for the entire composite. However, as the skill of the MRI operator increased, we were able to obtain a complete triad of landmarks in each stack with only a fine adjustment of the MRI field of view orientation.



#### **6.0.0.1 Description of data sets**

Table 6.0.0.1 shows the various studies done on each complete data set. There were two complete data sets on each of two subjects. They were intensively studied. The goal was to develop the algorithms and software to register MRI stacks to the neuromagnetometer PPN Headframe. The three techniques studied were:

##### **Direct**

The location of a cardinal landmark was estimated by picking pixel locations from the closest slice on the PixelPicker display subsystem of the CoordAligner. The results are compared with the recorded values from the Polhemus Navigation 3space digitizer probe.

##### **Pointer**

The location of a cardinal landmark was calculated by locating the blobs that comprised each spoke's trace and calculating the point of intersection of the pointer spoke's. The results are compared with the recorded values from the Polhemus digitizer probe.

##### **SpiderWeb Robot Spider**

The location of a cardinal landmark was determined by steering a software cursor, or "robot spider" on the SpiderWeb skin isosurface. The isosurface threshold value for skin was estimated by eye; a reasonable value would yield a clean surface. The location of the landmark is constrained to lie on the isosurface, and its position was determined by eye and comparison with photographs of the skin cardinal landmarks. The results are compared with the recorded values from the probe.

### **6.0.1 Results of Localization of the Cardinal Landmarks from the Skin**

An integral part of the Biomagnetic Technology, Inc., neuromagnetometer systems is a Polhemus Navigation 3Space digitizer probe. A standard procedure for every neuromagnetic study is the measurement of the subject head position. In this procedure the neuromagnetometer operator would touch the tip of the digitizer probe

to the landmarks on the skin of the subject. The Polhemus digitizer probe reports the position and orientation of its sensor probe in space relative to a transmitter coil via a 10 kHz electromagnetic field. The theory of operation is described by Raab *et al.* 1979[243]. Essentially, the transmitter consists of three orthogonal coils that alternately provide a burst of AC electromagnetic field. The probe also has three orthogonal coils, and the induced voltages in them are recorded. From the 9 induced voltages the position and orientation of the stylus relative to the transmitting coil can be accurately calculated by a dedicated microprocessor. This is due to the over-determined nature of problem, whereby the nine values of data are used to deduce the values of the six position and orientation parameters. The digitizer version of the probe is optimized for static, high precision recording of points near the transmitter. The orientation information of the probe is not used in these studies.

#### **6.0.1.1 Specifications for the Polhemus 3Space Digitizer Measurements**

The device has a nominal position tolerance of 0.081 cm RMS for the  $X$ ,  $Y$ , or  $Z$  probe position. It has a nominal resolution of 0.0041 cm. It can make measurements within this tolerance inside a box 40.0 cm  $\times$  40.0 cm  $\times$  25.4 cm surrounding the source coil. It can function with reduced accuracy within a 100 cm  $\times$  100 cm  $\times$  25.5 cm volume. It reports, via a serial RS-232 communications line, up to 60 times per to the host computer. The Polhemus digitizer is used with the BTI neuromagnetometer to determine the position of a subject, who wears a receiver on a head band, to track the relative position of the subject's head. A probe is used to register the locations of the cardinal landmarks so that the measured points can be expressed in the PPN Headframe. For our purposes, it provides the comparison reference for our subsequent measurements from MRI.

Given that the Polhemus is capable of being so precise, the next problem is to determine how accurately a trained operator can locate the cardinal landmarks. Repeated measurements were taken of each subject to determine the accuracy of locating a position on the skin.

#### **6.0.1.2 Head Measurement Procedure for the Neuromagnetometer**

The measurement procedure for registering a subjects head in the neuromagnetometer is as follows. The subject puts on a headband containing a Polhemus receiver coil.

The purpose of this is to track relative movement of the subject's head between neuromagnetometer recording sequences. However, the absolute position of the subject's head is not determined until careful digitization of the subject's cardinal landmarks. The subject lies down on the wooden neuromagnetometer bed, with the head resting on a conforming vacuum pillow that minimizes head movement without restraining the subject. The operator then, under computer prompting, touches the Polhemus probe to each of the subject's cardinal landmarks in turn. These probe measurements are repeated cyclically, from nasion, to left, then right periauricular point, twice. If the neuromagnetometer software determines that the standard deviation of the measurements are beyond accepted limits, the operator is prompted to retake the measurements. Before and after a recording session the Polhemus probe is used to determine the position of each landmark. Should the subject head have moved during the recording sessions, it would be revealed by the discrepancy between the before and after measurements.

### **6.0.1.3 Key to Polhemus digitizer probe measurement tables headings**

For each set of data, shown in Tables 6.2, 6.3, 6.4, and 6.5 cardinal landmark positions are averaged over repeated measurement procedures, described above, and expressed as PPN Headframe coordinates ( $X, Y, Z$ ) in centimeters. The values in parentheses are the standard deviations. The three cardinal landmarks, namely the left periauricular point, the right periauricular point, and nasion, were digitized in various subject/operator pairs. In addition, two additional locations, labeled Position 1 and Position 2, were digitized to provide an internal consistency check for precision. These are arbitrary scalp locations, and the standard deviation is an estimate of the precision for any arbitrary location in the PPN Headframe. The measurements are in centimeters, with the figures in parenthesis being the standard deviation of the measurements.

We show in Table 6.2 and 6.3 that the measurements of cardinal landmarks locations are extremely precise. The precision of repeated measures is better than 0.12 mm 6.3 Standard Deviation by an experienced operator (ZL) and 1.8 mm 6.5 Standard Deviation by an inexperienced operator (DBK), as evidenced by Table 6.5.



| Subject | Stack    | Direct    | Pointer    | Spider In Web |
|---------|----------|-----------|------------|---------------|
| SW      | Coronal  | Table 6.7 | Table 6.10 | Table 6.13    |
| SW      | Sagittal | Table 6.7 | Table 6.10 | Table 6.13    |
| SW      | Axial    | Table 6.7 | Table 6.10 | Table 6.13    |
| ZL      | Coronal  | Table 6.7 | Table 6.10 | Table 6.13    |
| ZL      | Sagittal | Table 6.7 | Table 6.10 | Table 6.13    |

Table 6.1: Overview of registration studies from acceptable MRI stacks.

| Landmark   | X     | <i>S.D.</i> | Y     | <i>S.D.</i> | Z     | <i>S.D.</i> |
|------------|-------|-------------|-------|-------------|-------|-------------|
| Nasion     | 11.0  | (0.01)      | 0.00  | ( NA )      | 0.00  | ( NA )      |
| LPA        | 0.25  | (0.12)      | 7.45  | (0.05)      | 0.00  | ( NA )      |
| RPA        | -0.25 | (0.23)      | -7.45 | (0.05)      | 0.00  | ( NA )      |
| Position 1 | 0.37  | (0.14)      | 0.75  | (0.09)      | 13.10 | (0.08)      |
| Position 2 | 3.42  | (0.09)      | -3.53 | (0.05)      | 11.69 | (0.06)      |

Table 6.2: Cardinal landmarks for subject SW, as measured by ZL, using a Polhemus probe, with 3 repetitions of 3 measurements procedures. Measurements in cm.

| Landmark | X     | <i>S.D.</i> | Y     | <i>S.D.</i> | Z    | <i>S.D.</i> |
|----------|-------|-------------|-------|-------------|------|-------------|
| Nasion   | 9.99  | (0.00)      | 0.00  | ( NA )      | 0.00 | ( NA )      |
| LPA      | 0.02  | (0.13)      | 7.25  | (0.04)      | 0.00 | ( NA )      |
| RPA      | -0.02 | (0.13)      | -7.25 | (0.04)      | 0.00 | ( NA )      |

Table 6.3: Cardinal landmarks for subject ZL as measured by SW, using a Polhemus probe, with 3 repetitions in each of 2 procedures and 2 repetitions for 1 procedure. Measurements in cm.

| Landmark   | X      | <i>S.D.</i> | Y     | <i>S.D.</i> | Z    | <i>S.D.</i> |
|------------|--------|-------------|-------|-------------|------|-------------|
| Nasion     | 10.4   | (0.00)      | 0.00  | ( NA )      | 0.00 | ( NA )      |
| LPA        | 0.096  | (0.11)      | 7.71  | (0.04)      | 0.00 | ( NA )      |
| RPA        | -0.096 | (0.11)      | -7.71 | (0.04)      | 0.00 | ( NA )      |
| Position 1 | -0.88  | (0.15)      | -0.18 | (0.11)      | 12.6 | (0.03)      |
| Position 2 | 5.69   | (0.10)      | -5.97 | (0.04)      | 6.84 | (0.10)      |

Table 6.4: Cardinal landmarks for subject DK measured by ZL, using a Polhemus probe, with 4 repetitions in each of 2 procedures. Measurements in cm.

| Landmark   | X     | <i>S.D.</i> | Y     | <i>S.D.</i> | Z    | <i>S.D.</i> |
|------------|-------|-------------|-------|-------------|------|-------------|
| Nasion     | 10.2  | (0.18)      | 0.00  | ( NA )      | 0.00 | ( NA )      |
| LPA        | -0.00 | (0.08)      | 7.24  | (0.12)      | 0.00 | ( NA )      |
| RPA        | 0.00  | (0.08)      | -7.24 | (0.12)      | 0.00 | ( NA )      |
| Position 1 | 0.09  | (0.07)      | 3.74  | (0.08)      | 5.50 | (0.09)      |
| Position 2 | -1.14 | (0.33)      | 4.09  | (0.11)      | 12.2 | (0.05)      |

Table 6.5: Cardinal landmarks for subject ZL, measured by DK, using a Polhemus probe, with 5 repetitions in 1 procedure, and 4 repetitions in 1 procedure. Measurements in cm.

## 6.0.2 Localization of Cardinal Landmarks in MRI Stacks

For all the subjects, slices, and stacks in this study, the base resolution figures for each data set are given in Table 6.6. Two data sets were included out of 4 MRI studies conducted. Two subjects that had acceptable MRI studies (SW and ZL), and two studies were rejected because of insufficient coverage of the cardinal landmarks to obtain registration cardinal landmarks.

The size of each pixel is given as the I size and the J size in cm. A pixel is a picture element, the MRI signal strength value at a point in the slice plane. The pixel is generally square, with a length (I size) and a breath (J size). In a 3D sense, each pixel can be considered a box  $1 \times 1 \times 3$  mm on a side. The pixel value represents a sample value for that volume. The number of pixels in each row of an image is the I resolution, and the number of pixels in each column is the J resolution. The number of pixels in a slice is I resolution  $\times$  J resolution. The size of each pixel was 0.09 cm on the I and J side, for a nominal pixel size of 1 mm. The distance between slices, or the K dimension, was 3 mm. There were  $256 \times 256$  pixels in a slice. The slice interval is the distance between adjacent slices. The slice factor is the fraction of the distance between slices sampled by the slice. A slice factor of 1.0 means that the slices were 3.0 mm thick, extending from the slice plane 1.5 mm on each side of the slice. All of the slices in these studies were 3.0 mm thick and had a 3 mm slice interval, and had a slice factor of 1.0.

| Subject | Stack    | I Size | I rez | J Size | J rez | Thick | K    |
|---------|----------|--------|-------|--------|-------|-------|------|
| SW      | Sagittal | 0.09   | 256   | 0.09   | 256   | 0.3   | 0.30 |
| SW      | Axial    | 0.09   | 256   | 0.09   | 256   | 0.3   | 0.30 |
| SW      | Coronal  | 0.09   | 256   | 0.09   | 256   | 0.3   | 0.30 |
| ZL      | Sagittal | 0.09   | 256   | 0.09   | 256   | 0.3   | 0.30 |
| ZL      | Coronal  | 0.09   | 256   | 0.09   | 256   | 0.3   | 0.30 |

Table 6.6: MRI Studies on the Philips Gyroscan 1.5 T system, with pixel and slice sizes expressed in cm.



#### **6.0.2.1 Results of Direct localization of Cardinal Landmarks from MRI Slices**

In the study of the localization of cardinal landmarks by the Direct Mode, an experienced CoordAligner operator (DK) selects the slice closest to the landmark. The operator selects a slice location associated with a landmark by using the PixelPicker slice window of the CoordAligner software. This procedure was carried out for each stack of image slices for each subject. This resulted in a total of 5 measurement sets on 2 subjects.

#### **6.0.2.2 Key to Direct slice measurements tables headings**

All of the three sets of three tables that follow have common headings. The **Subject** column identifies the person being studied. The **Stack** column indicates which slice orientation the group (stack) of slices was used from the MRI study. The **Cardinal** indicates the cardinal landmark located. The **X,Y,Z** column gives the coordinates in the PPN Headframe in cm.

| Subject | Stack    | Cardinal | X     | Y     | Z    |
|---------|----------|----------|-------|-------|------|
| Sub: SW | SAGITTAL |          |       |       |      |
| SW      | Sagittal | Nasion   | 12.00 | 0.00  | 0.00 |
| SW      | Sagittal | LPA      | 0.07  | 7.61  | 0.00 |
| SW      | Sagittal | RPA      | -0.07 | -7.61 | 0.00 |
| Sub: SW | AXIAL    |          |       |       |      |
| SW      | Axial    | Nasion   | 12.40 | 0.00  | 0.00 |
| SW      | Axial    | LPA      | -0.63 | 8.26  | 0.00 |
| SW      | Axial    | RPA      | 0.63  | -8.26 | 0.00 |
| Sub: SW | CORONAL  |          |       |       |      |
| SW      | Coronal  | Nasion   | 11.90 | 0.00  | 0.00 |
| SW      | Coronal  | LPA      | 0.31  | 8.15  | 0.00 |
| SW      | Coronal  | RPA      | -0.31 | -8.15 | 0.00 |
| Sub: ZL | SAGITTAL |          |       |       |      |
| ZL      | Sagittal | Nasion   | 10.70 | 0.00  | 0.00 |
| ZL      | Sagittal | LPA      | -0.29 | 7.34  | 0.00 |
| ZL      | Sagittal | RPA      | 0.29  | -7.34 | 0.00 |
| Sub: ZL | CORONAL  |          |       |       |      |
| ZL      | Coronal  | Nasion   | 10.70 | 0.00  | 0.00 |
| ZL      | Coronal  | LPA      | 0.05  | 7.14  | 0.00 |
| ZL      | Coronal  | RPA      | -0.05 | -7.14 | 0.00 |

Table 6.7: Direct Localization Coordinates of Cardinal Landmarks expressed in PPN Headframe. Measurements in cm.

### 6.0.2.3 Key to Direct error distance table headings

The second table of each group gives the main error statistic. This is the distance between the PPN Headframe as determined by the Polhemus digitizer probe and as determined by direct localization on the MRI slice. The distance is the discrepancy between each pair of homologous landmarks, and is given in cm. The average error distance between each pair of homologous landmarks is given in a separate row, labeled *AVERAGE*. This is also described as **Slop Error** distances in Section 5.4.2.1.



| Subject | Stack    | Landmark       | Distance |
|---------|----------|----------------|----------|
| SW      | Sagittal | Nasion         | 1.0      |
| SW      | Sagittal | LPA            | 0.24     |
| SW      | Sagittal | RPA            | 0.24     |
| SW      | Sagittal | <i>AVERAGE</i> | 0.49     |
| SW      | Axial    | Nasion         | 1.40     |
| SW      | Axial    | LPA            | 1.20     |
| SW      | Axial    | RPA            | 1.20     |
| SW      | Axial    | <i>AVERAGE</i> | 1.26     |
| SW      | Coronal  | Nasion         | 0.90     |
| SW      | Coronal  | LPA            | 0.70     |
| SW      | Coronal  | RPA            | 0.70     |
| SW      | Coronal  | <i>AVERAGE</i> | 0.77     |
| ZL      | Sagittal | Nasion         | 0.71     |
| ZL      | Sagittal | LPA            | 0.32     |
| ZL      | Sagittal | RPA            | 0.32     |
| ZL      | Sagittal | <i>AVERAGE</i> | 0.45     |
| ZL      | Coronal  | Nasion         | 0.71     |
| ZL      | Coronal  | LPA            | 0.11     |
| ZL      | Coronal  | RPA            | 0.11     |
| ZL      | Coronal  | <i>AVERAGE</i> | 0.31     |

Table 6.8: Direct localization *vs.* Polhemus digitizer probe measurement error distances in cm.

#### 6.0.2.4 Key to warpage table headings

The last table in each group gives the warpage of distances as measured by the Polhemus digitizer probe and direct slice measurement. The heading **LMark1** and **LMark2** give the name of the cardinal pair used to calculate the distances. The  $\Delta$  **Distance** is the error distance in cm. The **% Change** is the percentage change based on the Polhemus measured distance. The average scale change between the two measurement systems is shown in the row This is illustrated in 5.4.2.1.

| LMark 1        | LMark 2       | $\Delta$ Distance | % Change |
|----------------|---------------|-------------------|----------|
| LPA            | Nasion        | 1.07              | 8.2%     |
| RPA            | Nasion        | 0.78              | 5.7%     |
| RPA            | LPA           | 0.31              | 2.1%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.72              | 5.3%     |
| LPA            | Nasion        | 2.35              | 18.0%    |
| RPA            | Nasion        | 0.89              | 6.6%     |
| RPA            | LPA           | 1.66              | 11.1%    |
| <i>AVERAGE</i> | <i>CHANGE</i> | 1.63              | 11.9%    |
| LPA            | Nasion        | 1.09              | 8.3%     |
| RPA            | Nasion        | 1.19              | 8.8%     |
| RPA            | LPA           | 1.40              | 9.4%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 1.23              | 8.8%     |
| LPA            | Nasion        | 0.89              | 7.2%     |
| RPA            | Nasion        | 0.38              | 3.1%     |
| RPA            | LPA           | 0.19              | 1.3%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.49              | 3.9%     |
| LPA            | Nasion        | 0.49              | 4.0%     |
| RPA            | Nasion        | 0.55              | 4.4%     |
| RPA            | LPA           | -0.22             | -1.5%    |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.27              | 2.3%     |

Table 6.9: Direct method percentage change in distances between measurement modalities.



#### **6.0.2.5 Results of Pointer Localization of Cardinal Landmarks**

In this study, the CoordAlign program was used to reconstruct the position of the cardinal landmarks using the Pointers algorithm. The cardinal landmark location is obtained from the computed best estimate for where the spokes of the Pointer intersect. The in these runs, the Blobulator was not used to find the center of gravity of the blobby traces in each slice. The location of the blob centers were done by the PixelPicker in the CoordAlign operator by operator DK. The Pointers algorithm would find the best line and point given the blob locations provided by the operator. Errors in estimating the blob centers could be minimized by interactively minimizing the skew error by with the blob error. The Blobulator was not used to find blob centers because it tended to increase skew error. Appendix A Section A.4 gives a sample report detailing all of the component error in calculating the position of a landmark from the Pointers technique and algorithm.

| Subject  | Stack    | Cardinal | X     | Y     | Z    |
|----------|----------|----------|-------|-------|------|
| Sub: SW  | SAGITTAL |          |       |       |      |
| SW       | Sagittal | Nasion   | 11.70 | 0.00  | 0.00 |
| SW       | Sagittal | LPA      | 0.22  | 7.42  | 0.00 |
| SW       | Sagittal | RPA      | -0.22 | -7.42 | 0.00 |
| Sub: SW  | AXIAL    |          |       |       |      |
| SW       | Axial    | Nasion   | 11.20 | 0.00  | 0.00 |
| SW       | Axial    | LPA      | 0.18  | 7.46  | 0.00 |
| SW       | Axial    | RPA      | -0.18 | -7.46 | 0.00 |
| Sub: SW  | CORONAL  |          |       |       |      |
| SW       | Coronal  | Nasion   | 11.90 | 0.00  | 0.00 |
| SW       | Coronal  | LPA      | 0.18  | 7.51  | 0.00 |
| SW       | Coronal  | RPA      | -0.18 | -7.51 | 0.00 |
| Sub: ZL  | SAGITTAL |          |       |       |      |
| hline ZL | Sagittal | Nasion   | 10.00 | 0.00  | 0.00 |
| ZL       | Sagittal | LPA      | -0.02 | 7.25  | 0.00 |
| ZL       | Sagittal | RPA      | 0.02  | -7.25 | 0.00 |
| Sub: ZL  | CORONAL  |          |       |       |      |
| ZL       | Coronal  | Nasion   | 10.30 | 0.00  | 0.00 |
| ZL       | Coronal  | LPA      | 0.20  | 7.27  | 0.00 |
| ZL       | Coronal  | RPA      | -0.20 | -7.27 | 0.00 |

Table 6.10: Pointer Localization Coordinates of Cardinal Landmarks.

| Subject | Stack    | Landmark       | Distance |
|---------|----------|----------------|----------|
| SW      | Sagittal | Nasion         | 0.70     |
| SW      | Sagittal | LPA            | 0.04     |
| SW      | Sagittal | RPA            | 0.04     |
| SW      | Sagittal | <i>AVERAGE</i> | 0.26     |
| SW      | Axial    | Nasion         | 0.20     |
| SW      | Axial    | LPA            | 0.07     |
| SW      | Axial    | RPA            | 0.07     |
| SW      | Axial    | <i>AVERAGE</i> | 0.11     |
| SW      | Coronal  | Nasion         | 0.90     |
| SW      | Coronal  | LPA            | 0.09     |
| SW      | Coronal  | RPA            | 0.09     |
| SW      | Coronal  | <i>AVERAGE</i> | 0.36     |
| ZL      | Sagittal | Nasion         | 0.01     |
| ZL      | Sagittal | LPA            | 0.04     |
| ZL      | Sagittal | RPA            | 0.04     |
| ZL      | Sagittal | <i>AVERAGE</i> | 0.03     |
| ZL      | Coronal  | Nasion         | 0.31     |
| ZL      | Coronal  | LPA            | 0.18     |
| ZL      | Coronal  | RPA            | 0.18     |
| ZL      | Coronal  | <i>AVERAGE</i> | 0.22     |

Table 6.11: Pointer localization *vs* Polhemus digitization measurement error distances in cm.



| LMark 1        | LMark 2       | $\Delta$ Distance | % Change |
|----------------|---------------|-------------------|----------|
| LPA            | Nasion        | 0.59              | 4.5%     |
| RPA            | Nasion        | 0.55              | 4.1%     |
| RPA            | LPA           | -0.06             | -0.4%    |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.36              | 2.7%     |
| LPA            | Nasion        | 0.23              | 1.7%     |
| RPA            | Nasion        | 0.11              | 0.8%     |
| RPA            | LPA           | 0.02              | 0.1%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.12              | 0.9%     |
| LPA            | Nasion        | 0.84              | 6.4%     |
| RPA            | Nasion        | 0.73              | 5.4%     |
| RPA            | LPA           | 0.12              | 0.8%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.56              | 4.2%     |
| LPA            | Nasion        | 0.04              | 0.3%     |
| RPA            | Nasion        | -0.02             | -0.2%    |
| RPA            | LPA           | 0.00              | 0.0%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.00              | 0.0%     |
| LPA            | Nasion        | 0.12              | 0.9%     |
| RPA            | Nasion        | 0.41              | 3.3%     |
| RPA            | LPA           | 0.05              | 0.3%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.19              | 1.5%     |

Table 6.12: Pointer technique percentage change in distances between measurement modalities.

#### **6.0.2.6 Results of Localization of Cardinal Landmarks on SpiderWeb Skin Isosurface**

The goal of using the SpiderWeb skin isosurface is to simulate the geometry of the skin. Cardinal landmarks and other features can be identified, measured, and analyzed from this surface. By comparing the location of landmarks identified *in vivo* with skin isosurfaces, we can evaluate the fidelity of the skin surfaces, and the cardinal landmark locations. Figure 6.1 shows the nasion of subject SW from the and profile. Figure 6.5 show the left periauricular point. For subject ZL, we also show nasion from the front and profile. We can compare these photographs with the skin reconstructions from MRI stacks. The Pointers devices are occasionally reconstructed with the skin surface if the contrast media chambers came too close to the skin. Information from the Pointers was not used to guide the SpiderWeb bug as much as consciously possible by the operator (DK).





Figure 6.1: Subject SW nasion photograph, profile view.



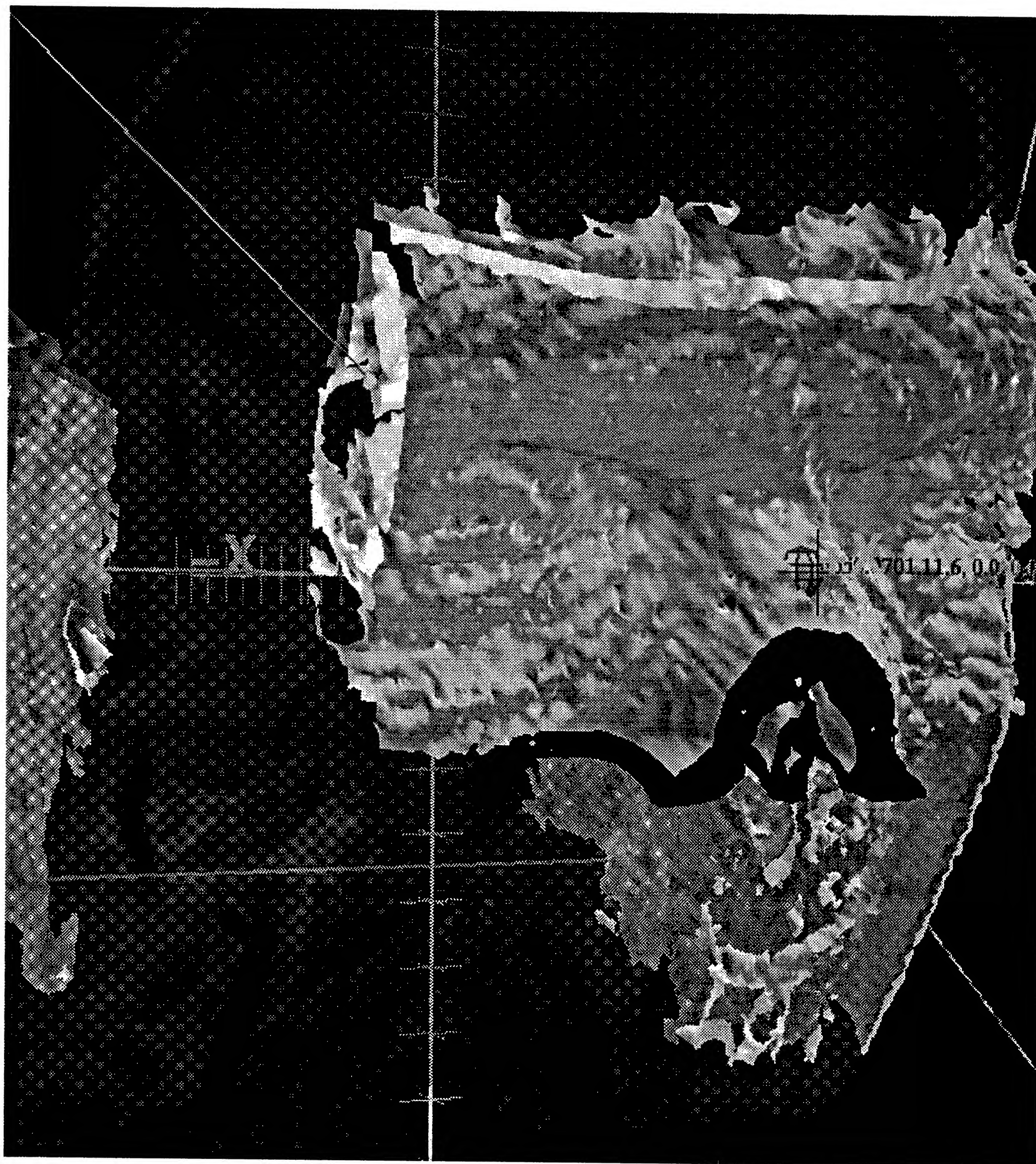


Figure 6.2: SW nasion as rendered from a coronal stack with the SpiderWeb algorithm.



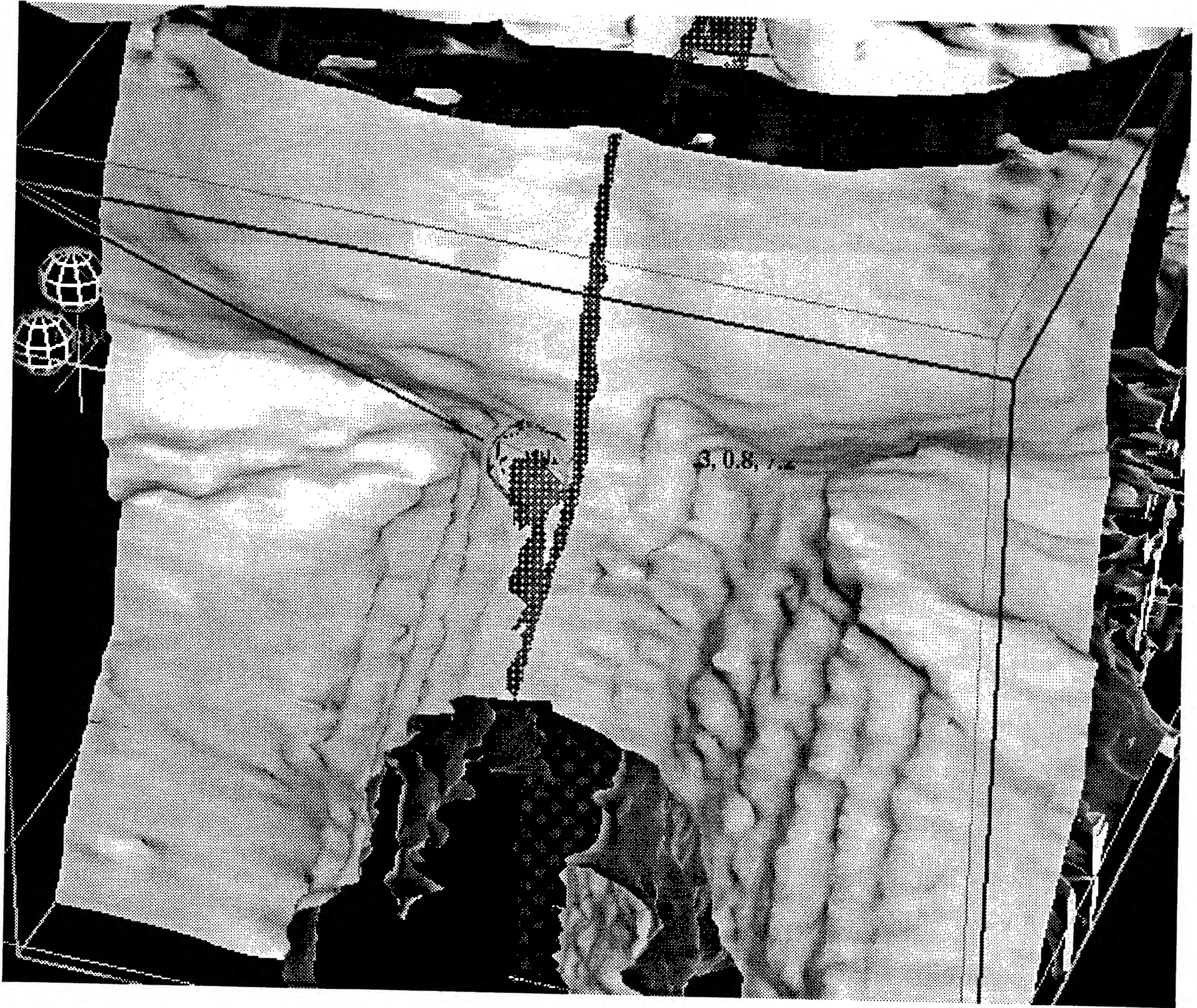


Figure 6.3: SW nasion as rendered from a sagittal stack with the SpiderWeb algorithm.



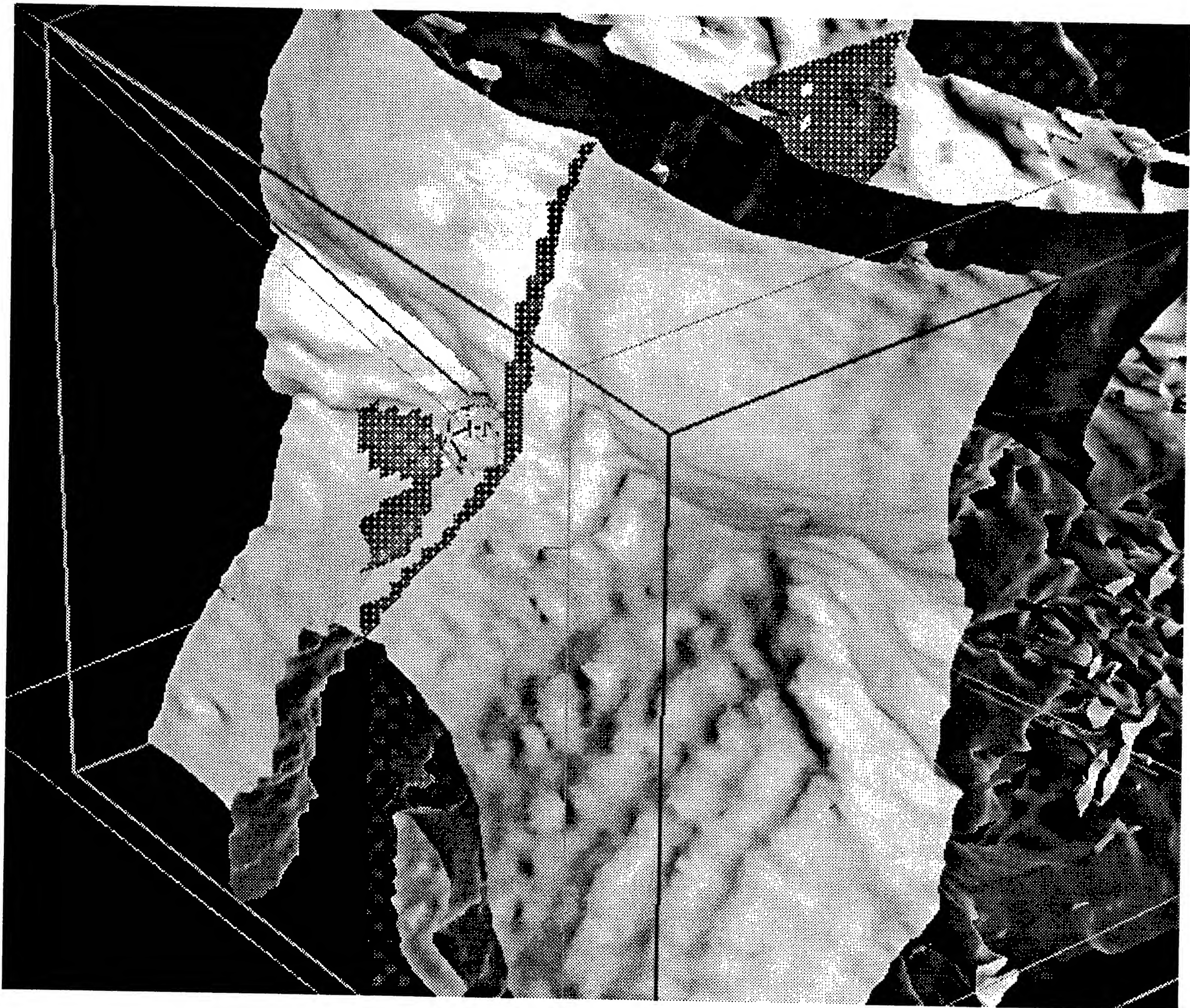


Figure 6.4: SW nasion, another view, as rendered from a sagittal stack with the SpiderWeb algorithm.





Figure 6.5: Subject SW left ear photograph.



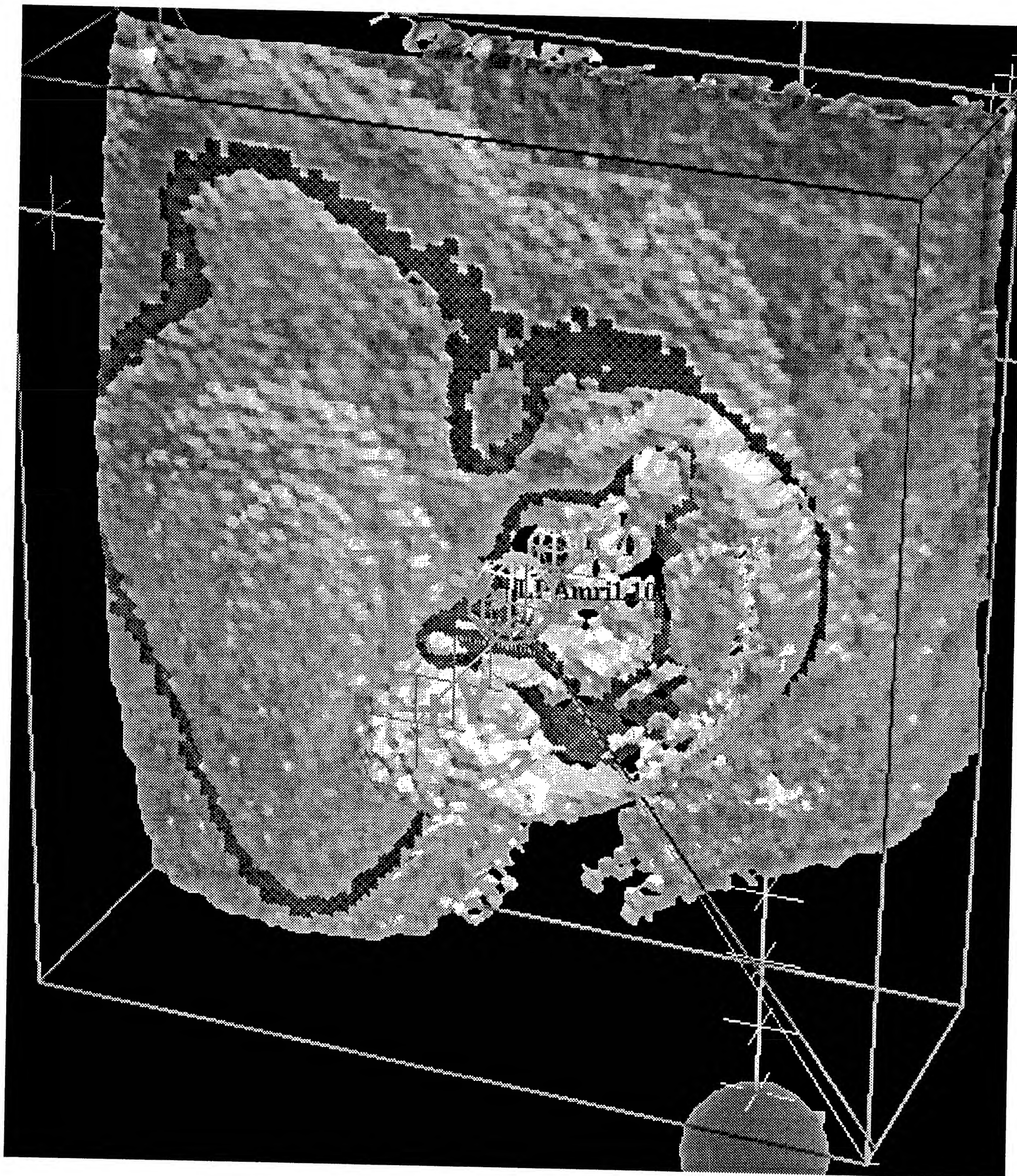


Figure 6.6: Subject SW left ear reconstruction from sagittal slices with left periauricular point marked.



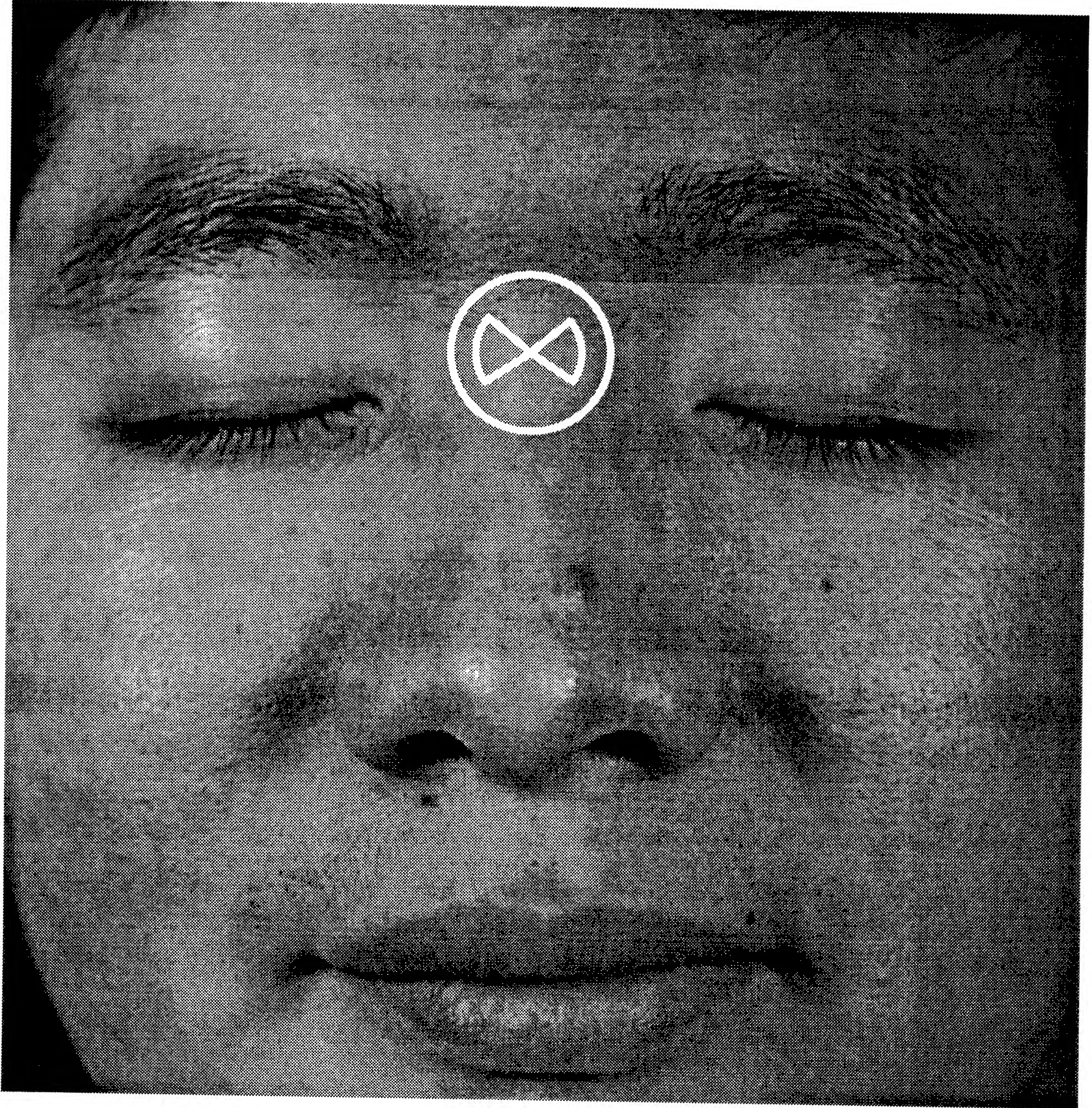


Figure 6.7: ZL's Frontal Face Photograph.



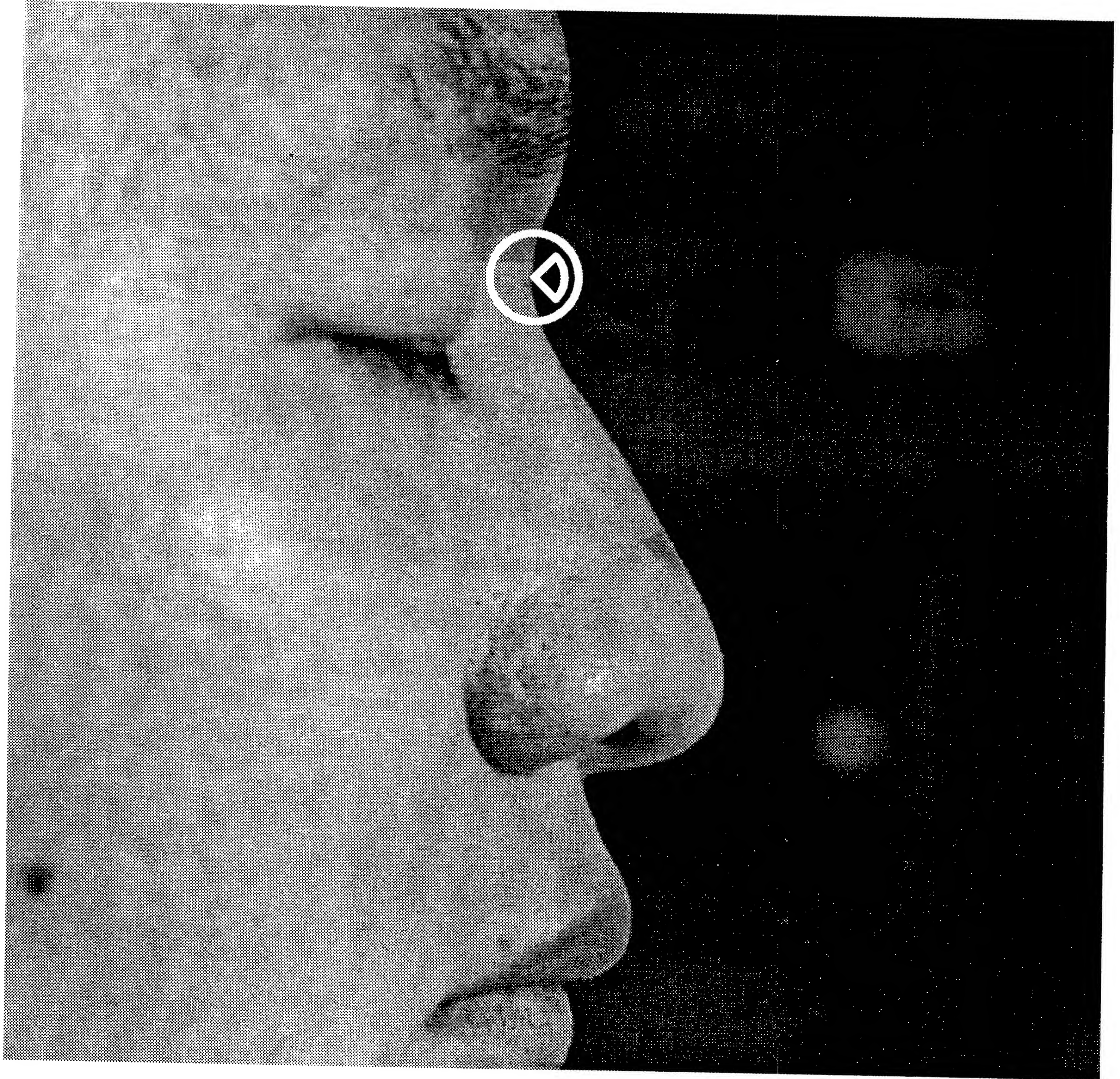


Figure 6.8: ZL Nasion Photograph.



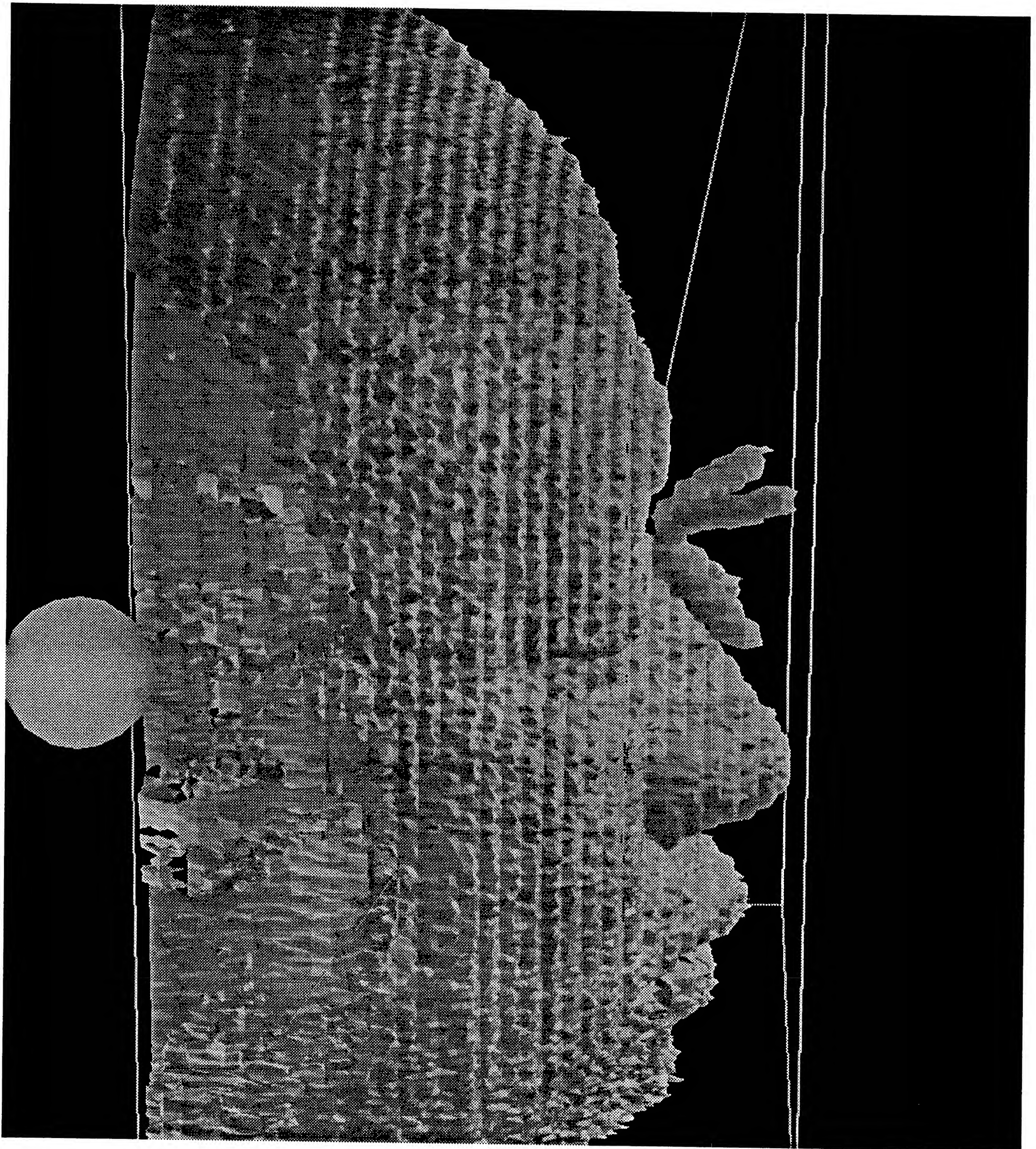


Figure 6.9: Subject ZL nasion reconstruction from a coronal stack.





Figure 6.10: Subject ZL nasion reconstructed from sagittal slices.



The following tables give the results of SpiderWeb bug surface localization of the cardinal landmarks on two subject (SW and ZL) taken from coronal, sagittal, and axial (SW) stacks from Philips GYROSCAN<sup>TM</sup> 1.5 T MRI system at the New York University Medical Center.

| Subject | Stack    | Cardinal | X     | Y     | Z    |
|---------|----------|----------|-------|-------|------|
| Sub: SW | SAGITTAL |          |       |       |      |
| SW      | Sagittal | Nasion   | 11.30 | 0.00  | 0.00 |
| SW      | Sagittal | LPA      | 0.46  | 7.82  | 0.00 |
| SW      | Sagittal | RPA      | -0.46 | -7.82 | 0.00 |
| Sub: SW | AXIAL    |          |       |       |      |
| SW      | Axial    | Nasion   | 11.30 | 0.00  | 0.00 |
| SW      | Axial    | LPA      | -0.06 | 7.62  | 0.00 |
| SW      | Axial    | RPA      | 0.06  | -7.62 | 0.00 |
| Sub: SW | CORONAL  |          |       |       |      |
| SW      | Coronal  | Nasion   | 11.20 | 0.00  | 0.00 |
| SW      | Coronal  | LPA      | 0.19  | 7.52  | 0.00 |
| SW      | Coronal  | RPA      | -0.19 | -7.52 | 0.00 |
| Sub: ZL | SAGITTAL |          |       |       |      |
| ZL      | Sagittal | Nasion   | 10.20 | 0.00  | 0.00 |
| ZL      | Sagittal | LPA      | -0.05 | 7.38  | 0.00 |
| ZL      | Sagittal | RPA      | 0.05  | -7.38 | 0.00 |
| Sub: ZL | CORONAL  |          |       |       |      |
| ZL      | Coronal  | Nasion   | 10.30 | 0.00  | 0.00 |
| ZL      | Coronal  | LPA      | 0.10  | 7.23  | 0.00 |
| ZL      | Coronal  | RPA      | -0.10 | -7.23 | 0.00 |

Table 6.13: SpiderWeb Localization Coordinates of Cardinal Landmarks. Values in cm.

| Subject | Stack    | Landmark       | Distance |
|---------|----------|----------------|----------|
| SW      | Sagittal | Nasion         | 0.30     |
| SW      | Sagittal | LPA            | 0.43     |
| SW      | Sagittal | RPA            | 0.43     |
| SW      | Sagittal | <i>AVERAGE</i> | 0.38     |
| SW      | Axial    | Nasion         | 0.30     |
| SW      | Axial    | LPA            | 0.35     |
| SW      | Axial    | RPA            | 0.35     |
| SW      | Axial    | <i>AVERAGE</i> | 0.34     |
| SW      | Coronal  | Nasion         | 0.20     |
| SW      | Coronal  | LPA            | 0.09     |
| SW      | Coronal  | RPA            | 0.09     |
| SW      | Coronal  | <i>AVERAGE</i> | 0.13     |
| ZL      | Sagittal | Nasion         | 0.21     |
| ZL      | Sagittal | LPA            | 0.15     |
| ZL      | Sagittal | RPA            | 0.15     |
| ZL      | Sagittal | <i>AVERAGE</i> | 0.17     |
| ZL      | Coronal  | Nasion         | 0.31     |
| ZL      | Coronal  | LPA            | 0.08     |
| ZL      | Coronal  | RPA            | 0.08     |
| ZL      | Coronal  | <i>AVERAGE</i> | 0.16     |

Table 6.14: SpiderWeb localization *vs.* Polhemus digitization measurement error distances. Distances in cm.



| LMark 1        | LMark 2       | $\Delta$ Distance | % Change |
|----------------|---------------|-------------------|----------|
| LPA            | Nasion        | 0.29              | 2.2%     |
| RPA            | Nasion        | 0.63              | 4.7%     |
| RPA            | LPA           | 0.76              | 5.1%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.56              | 4.0%     |
| LPA            | Nasion        | 0.60              | 4.6%     |
| RPA            | Nasion        | 0.09              | 0.6%     |
| RPA            | LPA           | 0.33              | 2.2%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.34              | 2.5%     |
| LPA            | Nasion        | 0.25              | 1.9%     |
| RPA            | Nasion        | 0.16              | 1.2%     |
| RPA            | LPA           | 0.14              | 0.9%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.18              | 1.3%     |
| LPA            | Nasion        | 0.30              | 2.5%     |
| RPA            | Nasion        | 0.19              | 1.5%     |
| RPA            | LPA           | 0.26              | 1.8%     |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.25              | 1.9%     |
| LPA            | Nasion        | 0.18              | 1.4%     |
| RPA            | Nasion        | 0.31              | 2.5%     |
| RPA            | LPA           | -0.04             | -0.3%    |
| <i>AVERAGE</i> | <i>CHANGE</i> | 0.15              | 1.2%     |

Table 6.15: SpiderWeb technique percentage change in distances *vs.* Polhemus probe measurements. Distances in cm.



# Chapter 7

## Discussion

We have compared three modalities of measurement by measuring the same cardinal landmarks in each modality. The results agree within the tolerance of the MRI measurements,  $1\text{ mm} \times 1\text{ mm} \times 3\text{ mm}$ .

If the practical limitations of each technique can be overcome, each method has significant utility for registering MRI to other measurement modalities.

The limitations of the direct mode measurement are:

- Expert knowledge of anatomy required to get accurate locations.
- Locations limited to slice plane.

Advantages are:

- No hardware required, suitable for retrospective studies where no fiducial markers are available.

The limitations of the Pointers technique are:



- Placement limited by clearance about the head in the MRI bore or with helmet like head coils.

Advantages of the Pointers technique are:

- Precision without expert anatomic insight on the part of the operator possible.

The limitations of the SpiderWeb bug localization technique are:

- Anatomic knowledge required to locate seed for Spider to spin surface, and to steer robot Spider to landmark location.
- Vast computer memory and fast computer required for interactive bug localization.

The advantages of the SpiderWeb bug localization technique are:

- Can be run on MRI studies without fiducial markers.

#### 7.0.2.7 Summary of average measurement errors: Key to table

Table 7.1 gives summary statistics for all of the measurements in Section 6. The *AVERAGE* Slop, Warp, and Warp % from the preceding tables are assembled in this table. Heading **Sub** refers to the subject of the study. **Technique** refers to the instrumentality used to measure the cardinal landmarks. The **Direct**, **Pointers** and **SpiderWeb** technique were used to measure the three cardinal landmarks, as previously described. The error figures on these measurements give an indication of the quality of the measurements with respect to the Polhemus digitizer measurements. The **Slop** column heading refers to the average of the three error distances for all three cardinal landmarks, as described in 5.4.2.1. The **Warp** heading refers to the average warpage of the triangle legs, as described in Section 5.4.2.1. **Warp %** refers to the warpage as a percentage of the base of the Polhemus digitizer measurements. The **Wob X** and **Wob Y** refer to the X and Y Wobble, respectively. This is the difference in the X and Y coordinates, respectively, of the measurement pairs. It is described in Section 5.4.2.1.

| Sub | Technique | Stack    | Slop | Warp | Warp % | Wob X | Wob Y |
|-----|-----------|----------|------|------|--------|-------|-------|
| SW  | Direct    | Sagittal | 0.49 | 0.72 | 5.3%   | -0.18 | 0.16  |
| SW  | Direct    | Axial    | 1.26 | 1.63 | 11.9%  | -0.88 | 0.81  |
| SW  | Direct    | Coronal  | 0.77 | 1.23 | 8.8%   | 0.06  | 0.70  |
| ZL  | Direct    | Sagittal | 0.45 | 0.49 | 3.9%   | -0.31 | 0.09  |
| ZL  | Direct    | Coronal  | 0.31 | 0.27 | 2.3%   | 0.03  | -0.11 |
| SW  | Pointers  | Sagittal | 0.26 | 0.36 | 2.7%   | -0.03 | -0.03 |
| SW  | Pointers  | Axial    | 0.11 | 0.12 | 0.9%   | -0.07 | 0.01  |
| SW  | Pointers  | Coronal  | 0.36 | 0.56 | 4.2%   | -0.07 | 0.06  |
| ZL  | Pointers  | Sagittal | 0.03 | 0.00 | 0.0%   | -0.04 | 0.00  |
| ZL  | Pointers  | Coronal  | 0.22 | 0.19 | 1.5%   | 0.18  | 0.02  |
| SW  | SpiderWeb | Sagittal | 0.38 | 0.56 | 4.0%   | 0.21  | 0.37  |
| SW  | SpiderWeb | Axial    | 0.34 | 0.34 | 2.5%   | -0.31 | 0.17  |
| SW  | SpiderWeb | Coronal  | 0.13 | 0.18 | 1.3%   | -0.06 | 0.07  |
| ZL  | SpiderWeb | Sagittal | 0.17 | 0.25 | 1.9%   | -0.07 | 0.13  |
| ZL  | SpiderWeb | Coronal  | 0.16 | 0.15 | 1.2%   | 0.08  | -0.02 |

Table 7.1: Summary Error Slop, Warpage, and Wobble compared across techniques.

#### 7.0.2.8 Discussion of Summary Error Table

The results summarized in Table 7.1 reveal certain properties of the different measurement sets. First, we can see that the Direct measurement technique in general has the largest overall errors. Secondly, the best overall measurement are obtained with the Pointers technique. The SpiderWeb gives overall errors worse than the Pointers, but not as bad as the Direct method.

The worst errors occur for the Pointers in the Coronal orientation stack. This can be expected because the direction of least resolution is oriented perpendicular to the PPN Headframe  $Z = 0$  plane. The least errors occur for the Pointers in the axial orientation stack. The axial stack is aligned with the Headframe plane, and this is the orientation that this coordinate system is least sensitive to errors.

Overall, any error greater than the resolution of the MRI resolution are unacceptable. Acceptable errors are less than the MRI resolution. However, the MRI resolution is anisotropic. Therefore a strict cutoff at 1 mm or 3 mm is not possible. We have shown in the above table that the average errors are at the average resolution of the MRI system, better than 1 mm for measurements in axial stacks, and better than 3 mm for coronal or sagittal stacks.

In conclusion, we can measure cardinal landmark locations within the maximum resolution of the MRI recording system. In this study,  $> 1$  mm minimal nominal precision was obtained for Pointers measurements in stacks of slices.



## **Part IV**

# **Neuromagnetic Brain Function Studies**



## Chapter 8

# Visualizing and Interpreting Neuromagnetic Data

### 8.1 Sphere Fitting of the Head

#### 8.1.1 Introduction: Why fit a sphere inside the brain case ?

Magnetic fields measured outside the head arise from two classes of electric currents: “primary” current that flows in active tissue and “secondary” that represents the current flowing in the surrounding conducting medium. To deal with the secondary current in detail requires a numerical model for the conducting properties of the head. However, Hari and Ilmoniemi 1986 [244] pointed out that in most cases an adequate model is provided by a sphere of uniform conductivity provided its center is located on the center of curvature of the inner surface of the skull just above the active tissue. For this reason the occipital pole of the head can be modeled by a best fitting sphere for neuromagnetic measurements with little loss of accuracy.

The formula for a sphere centered at  $X_c, Y_c, Z_c$  with radius  $R$  is

$$(8.1) \quad (X - X_c)^2 + (Y - Y_c)^2 + (Z - Z_c)^2 = R^2$$

We are given a list of points, our problem is to find the sphere that best fits the list of points. In other words, the coefficients  $X_c, Y_c, Z_c, R$  what will minimize the sum



of the distances between each point  $(X_i, Y_i, Z_i)$  and the surface of the sphere. The overall statement of the problem is:

$$\left( \begin{bmatrix} X_0 & Y_0 & Z_0 \\ X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix} - \begin{bmatrix} X_c & Y_c & Z_c \end{bmatrix} \right)^2 = R$$

We solve the problem with a classic numerical algorithm, the amoeba simplex algorithm, using the Numerical Recipes C code library from Press *et al.* 1988 [204]. We know that a reasonable seed value for the best sphere  $(X_c, Y_c, Z_c)$  is in the vicinity of the origin  $(0, 0, 0)$  of head-frame coordinate system. We know the average diameter of a head is about 24 cm, so a reasonable seed value for  $R$  is 12 cm. The amoeba algorithm converges quite rapidly ( $< 100$  iterations) to an optimal solution for a large number ( $n < 100$ ) of points to fit.

Estimates of the best sphere and ellipse fitting a collection of points is a useful facility for biological measurements, and is included in a number of biometric computer packages. The main geometric assumption is that for the occipital region of the brain, the sphere is a satisfactory simplifying shape. The main problem is to find the best sphere that can be fitted inside the skull in the vicinity of the occiput.

Figure 8.1 shows the best sphere that fits inside the brain case, or skull. The size and position of the sphere is determined by an interactive facility of the CoordAligner software program. The program user selects a number of points in the CoordAligner PixelPicker slice window that represent the border of the brain case. Representative points are taken from a number of slices, generally 20 or 30 points in each slice. A good representation of the curvature can be obtained from 25 slices. These slice locations are transformed into the PPN head-based coordinate system. This list of coordinates is handed to the BestSphere subroutine, which returns with the center  $(X_c, Y_c, Z_c)$  and radius  $R$  of a best sphere that is outlined by the cloud of points. The C language source code is in the Appendix Section A.3.



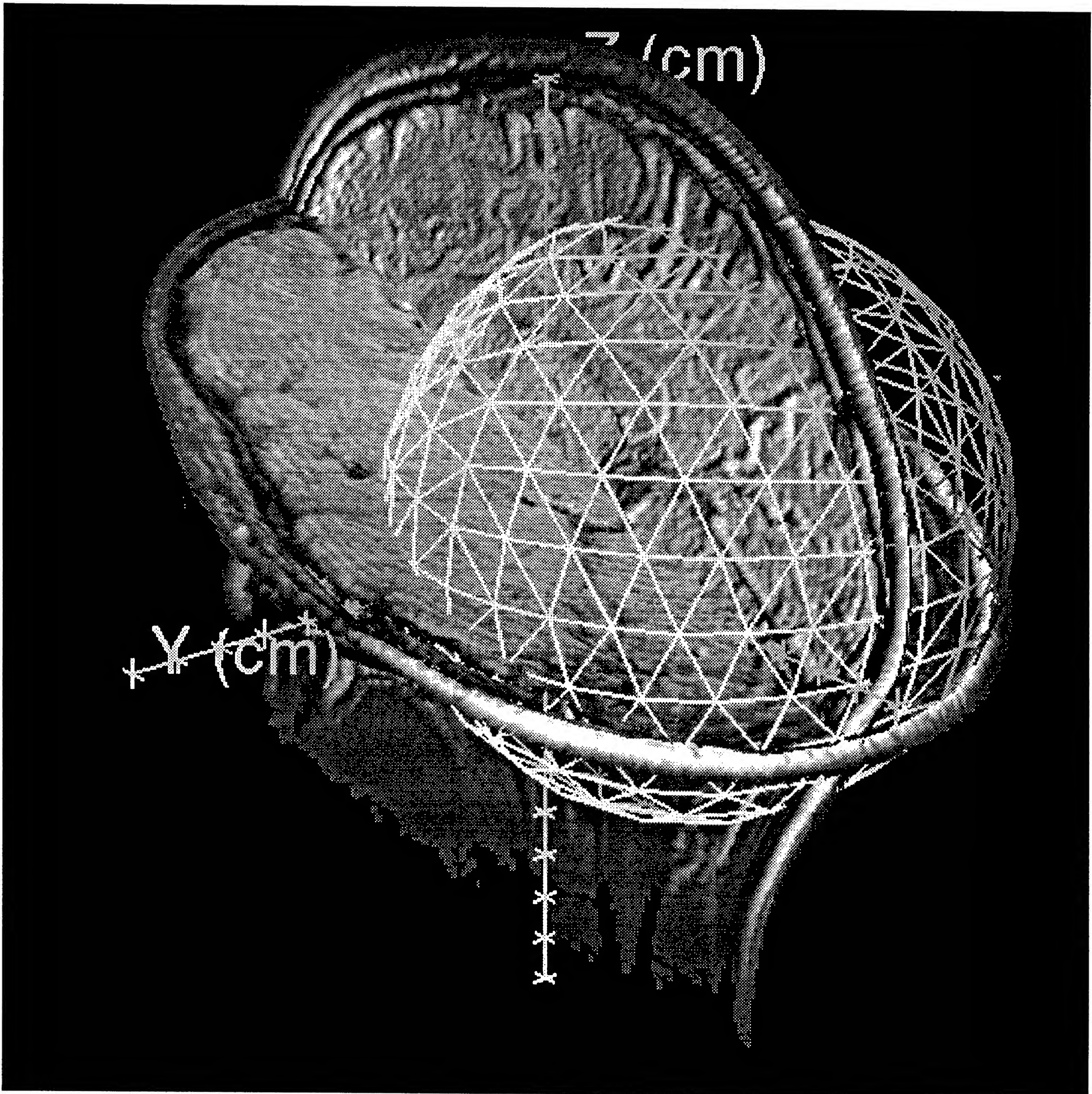


Figure 8.1: The best sphere that fits into the occipital pole of the brain case.



## 8.2 The magnetic skullcap: a 3D Delaunay triangularization of MEG data for visualization and modeling

An important first step in the interpretation of biomagnetic data is to visualize the raw and averaged data as it is recorded at the squid coils, before it is intensively smoothed, filtered, and otherwise processed. The most simple technique is to display on a monitor oscilloscope, the incoming waveforms from each squid in real time. The main problem with this is that it is difficult to visualize the spatial information that is the main feature of MEG. The head is magnetically transparent, so signals from sources are not refracted and spread by the volume conductor properties of the head. The magnetic skullcap represents the incoming magnetic field from each independent squid position as a tiled surface sheet, with intermediate values interpolated across the face of the constituent triangles.

Essentially, our approach to creating a measurement surface is to:

- Find the best sphere that fits the data points.
- Convert the data points in  $x, y, z$  to spherical coordinates  $\theta, \psi, \rho$  centered on the origin ( $\rho = 0$ ) of the best fitting sphere.
- Do the Delaunay triangularization on the  $\theta, \psi$  vertex coordinates. Take the list of edges and use the native vertices to build the edges.

The net effect is to build a planar Delaunay triangularization on the surface of the sphere, but knit the triangle with the different  $\rho$  values. The resultant surface is independent of the sphere surface. This surface is knit from the original SQUID coil measurement locations, and it not interpolated in any way onto a regular sphere. The only result from the Delaunay triangularization is that a list of simplicial complex (triangles) edges is found for the spatial locations of the SQUID measurement coils. They require a manifold “measurement surface” constructed from possibly irregularly sampled locations above the scalp. This is a true, manifold surface knit from irregular measurement locations without projection or interpolation, in the sense required by Wang and Kaufman 1992 [84][83]. This is a distinctly different algorithm from that used by Bruno and Williamson 1992 [245]. The measurement



locations are not mapped or distorted to lie on the surface of a sphere, and isocontours are not generated from already interpolated data. The isovalue contour lines are generated on the Delaunay triangle tiles using the SpiderWeb 2D algorithm. The scientific value of this approach is that once a surface is constructed, it is suitable for further analysis.

This algorithm is important because there is no interpolation of measurement locations, and no interpolation artifacts inherent in sparse, irregular data samples. There is no need to transform the spherical coordinate back to cartesian coordinates because the output of the Delaunay triangularization is lists of vertex triads to make into triangles. We simply array the  $x, y, z$  data into a new array of  $\theta, \psi, \rho$  values. A particular point  $x, y, z$  has the same array index  $i$  as the transformed coordinate  $\theta, \psi, \rho$ . A list of Delaunay edges consists of indices to triangle vertices. These indices can also be used to refer to  $x, y, z$  values, without further coordinate transformation.

The algorithm used to create a unique, minimal tiling of the irregular data collection points is known as a Delaunay triangularization[246]. Figure 8.2 shows a Voronoi diagram, with the Delaunay triangles shown in dashed lines. The Delaunay triangularization is the dual of Voronoi diagrams. Voronoi diagrams are systems of lines that are maximally distant from a set of points. Given some number of points in the plane, their Voronoi diagram divides the planes according to the nearest-neighbor rule. Specifically, that rule associates the region of the plane with that point. The Delaunay triangularization is an extension of the Voronoi region. A line in a Voronoi diagram is the border between two regions in the plane that are “owned” by the points. A Voronoi line represents a boundary between point-defined regions of the plane. An example would be the region served by a local post office. The neighborhood closest to each post office should be included in that delivery district. The boundary between districts is the Voronoi diagram. The Delaunay triangularization is a system of lines between each vertex. It contains an edge connecting two vertices in the plane if and only if their Voronoi regions share a common edge. More exactly, it is the set of all triangles formed by the sites such that the circumcircle of each triangle is empty of other sites. The set of edges of these triangles gives the Delaunay triangularization of the sites. Intuitively, the set of Delaunay triangles are the minimal triangular tiling of the surface. No triangle edges will cross any others, and each triangle is the smallest that can be tiled. The tiling is unique, there is no other tiling that satisfies all the requirements. Figure 8.5 shows a Voronoi decomposition of the measurement vertices. The lines extending out to infinity are not shown, and would not have the similar meaning on a sphere. The more interesting results are shown on Figure 8.6. By inspection, we can see the region “owned” by each vertex is carved up by the

Voronoi line segments. Here the Delaunay “hair-net” can be seen. It is interesting to take any of the measurement vertices, shown as triangulated squares, and find any vertex.

We used a Delaunay triangularization to minimize interpolation of data points. A Voronoi decomposition of the surface would give open polygons extending to infinity. On a spherical shell these do not have any meaning. We want to build a closed three dimensional surface. There is no known algorithm to build a true three dimensional Voronoi diagram or Delaunay triangularization. Indeed, the problem of tetrahedralization and triangularization tackled by Frey [87] to find a minimal tessellation of the cube is an attempt to find a three dimensional tetrahedral extension of Delaunay triangularization. We obtained a three dimensional Delaunay triangularization by projecting the measurement points on to a best-fit spherical surface(cf.8.1). From the resultant edge list, we used the original measurement vertices and generated a three dimensional mesh surface.

Consider the situation of a single, short, electric current near the surface of the scalp. What magnetic field is generated, and how would this magnetic field appear at the sensor surface ?

Figure 8.7 shows the 3d Delaunay triangularization of a number of arbitrary measurement locations about the back of the head. Figure 8.8 shows a side view of the head. These locations are knitted together into a surface, and the triangles colored by the intensity of the averaged magnetic field. The data was recorded on a BTi 5 sensor neuromagnetometer system. Isocontours show the magnetic field intensity at every 20 fT.

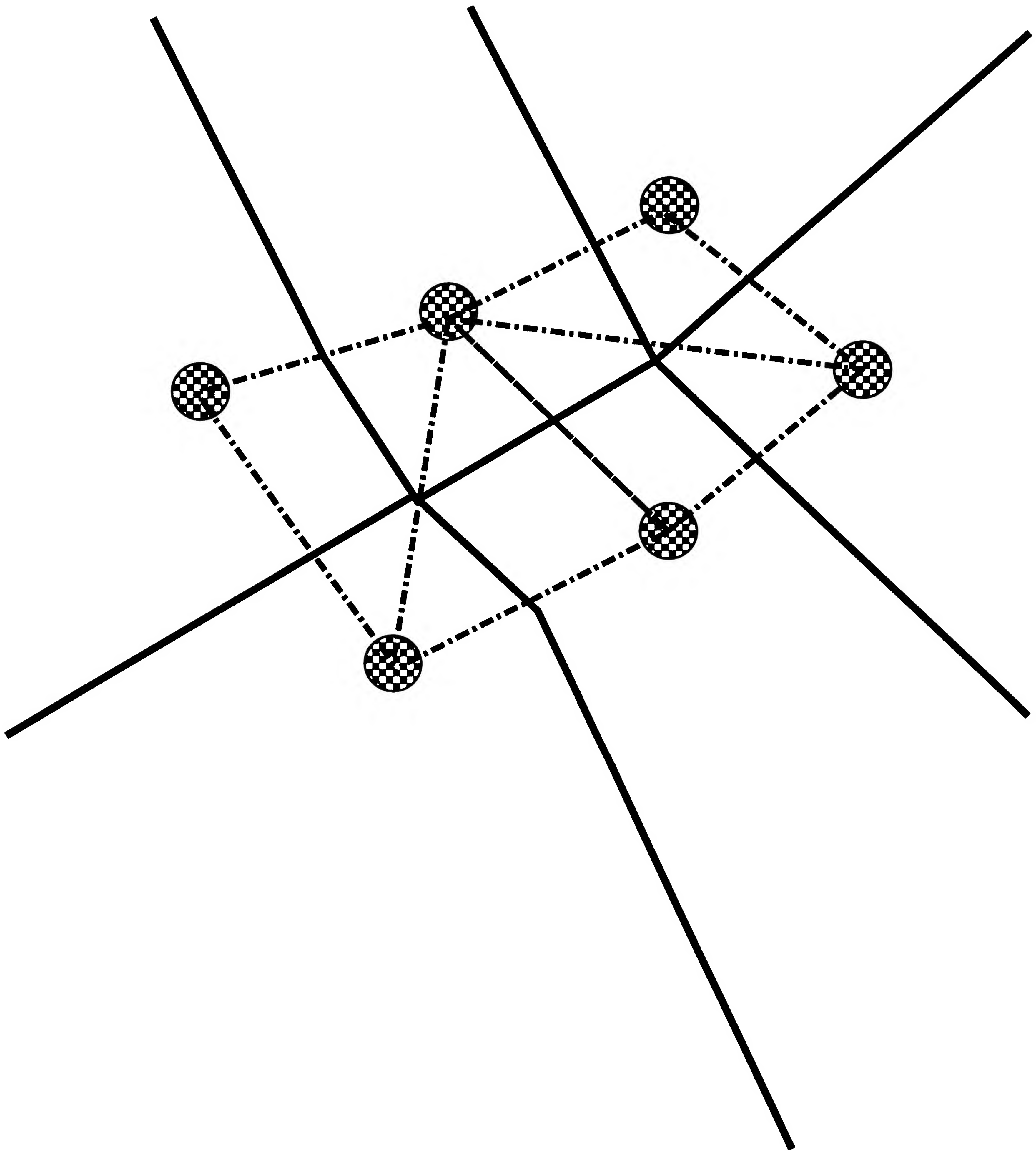


Figure 8.2: Voronoi diagram and Delaunay Triangularization in the plane.



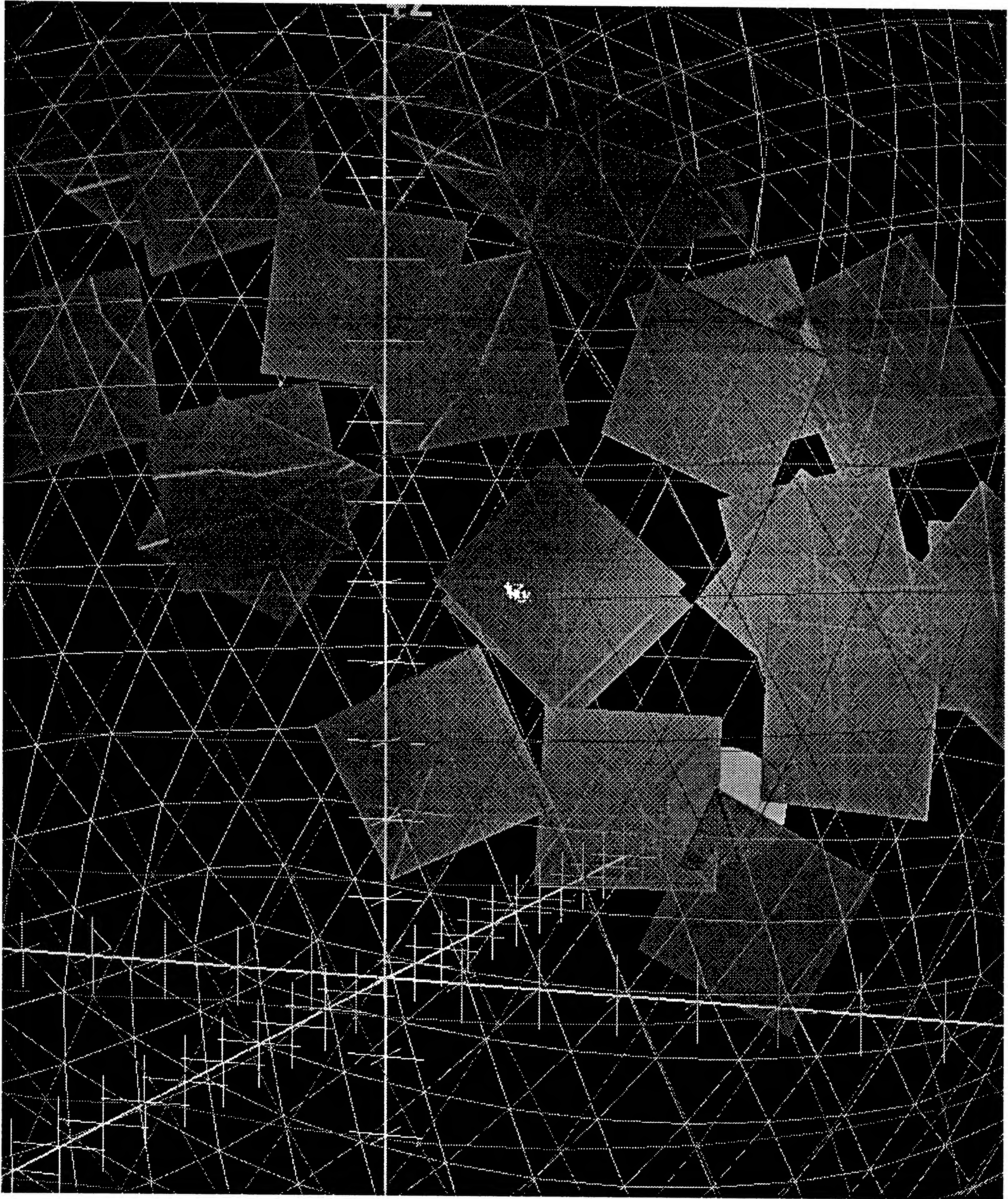


Figure 8.3: Best sphere and SQUID coils before Delaunay triangularization.



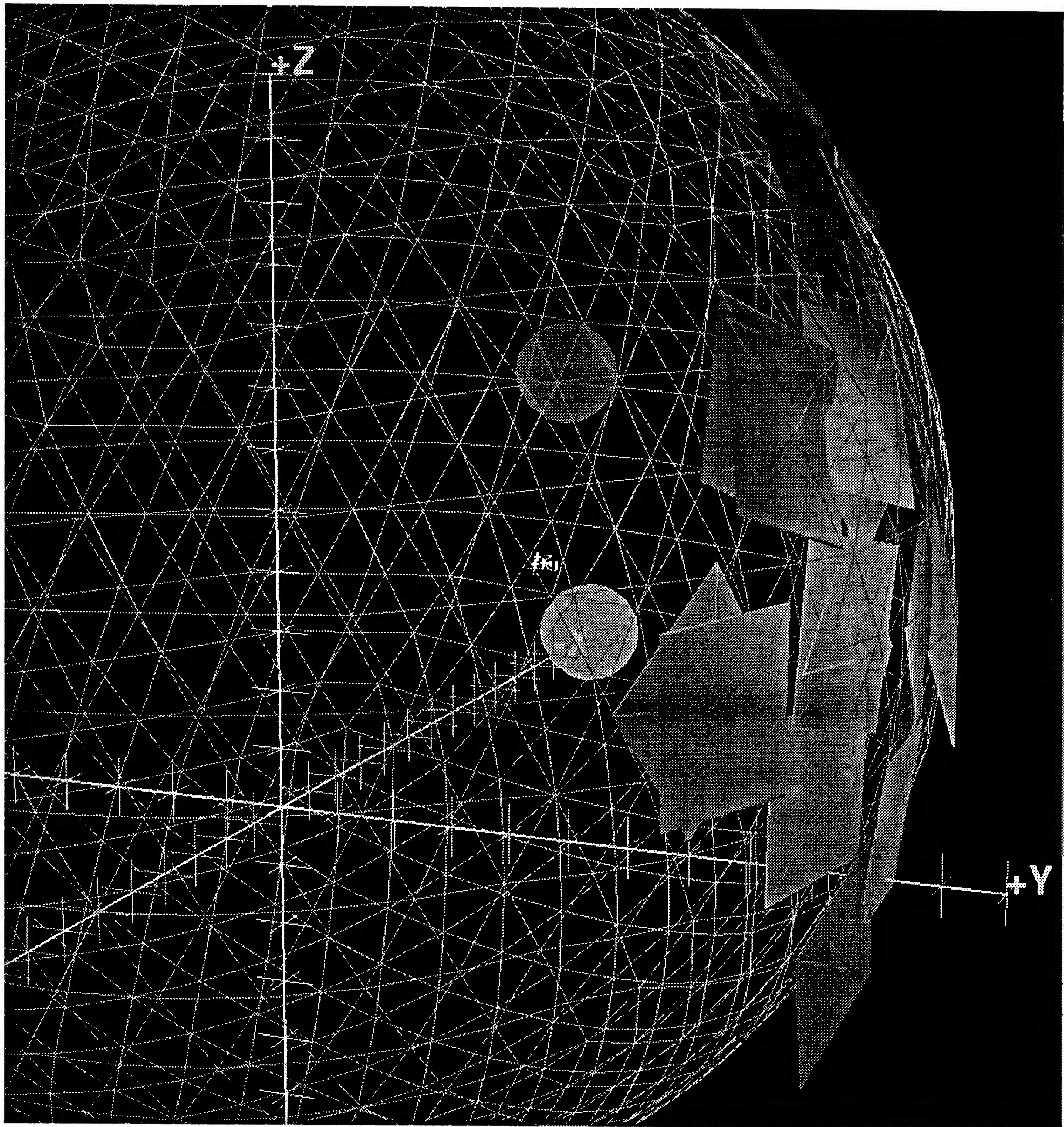


Figure 8.4: Side view best sphere and SQUID coils before Delaunay triangularization.



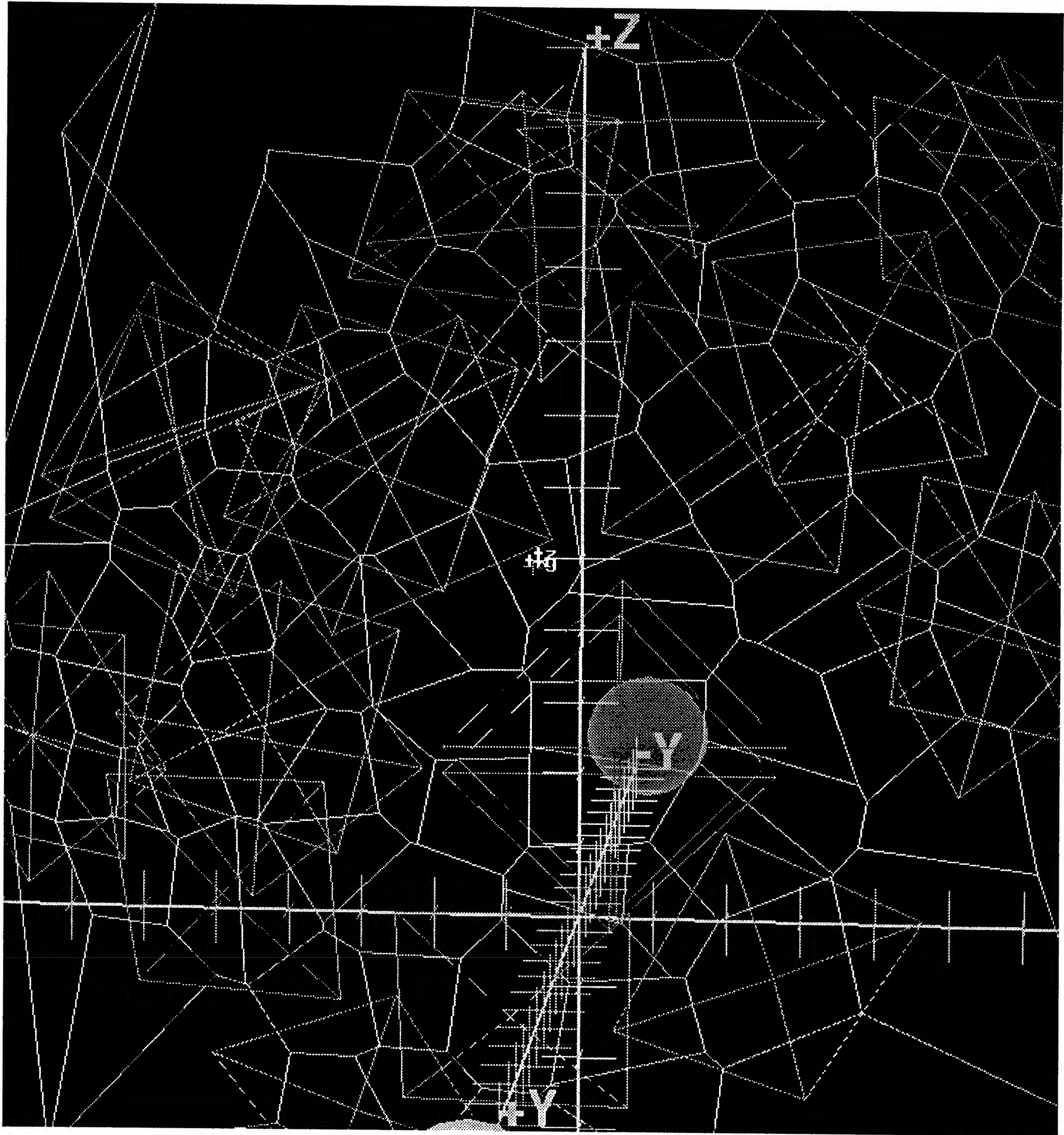


Figure 8.5: Voronoi surface of SQUID coil pickup.



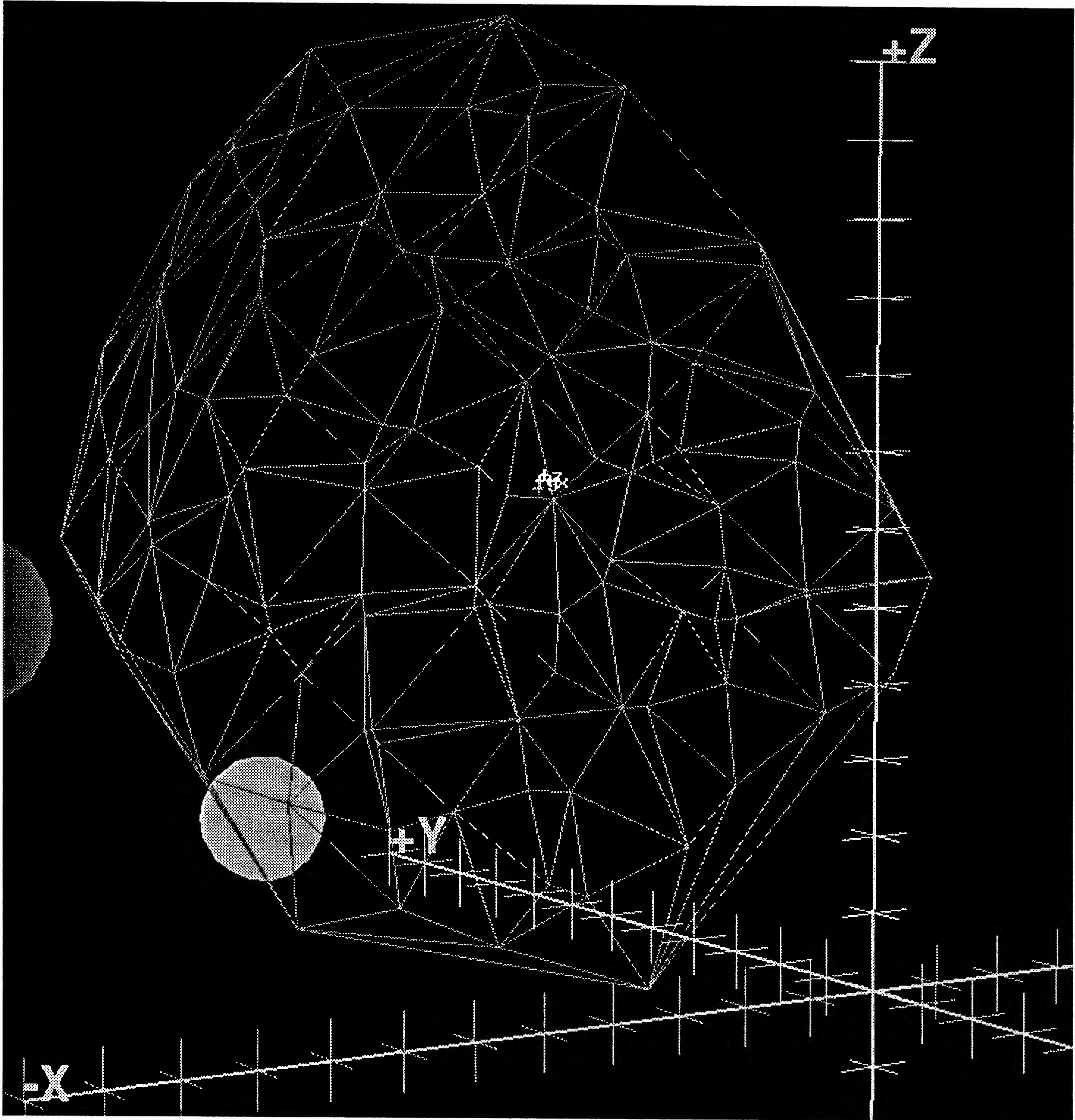


Figure 8.6: Delaunay triangular tileing of the SQUID coil pickups.



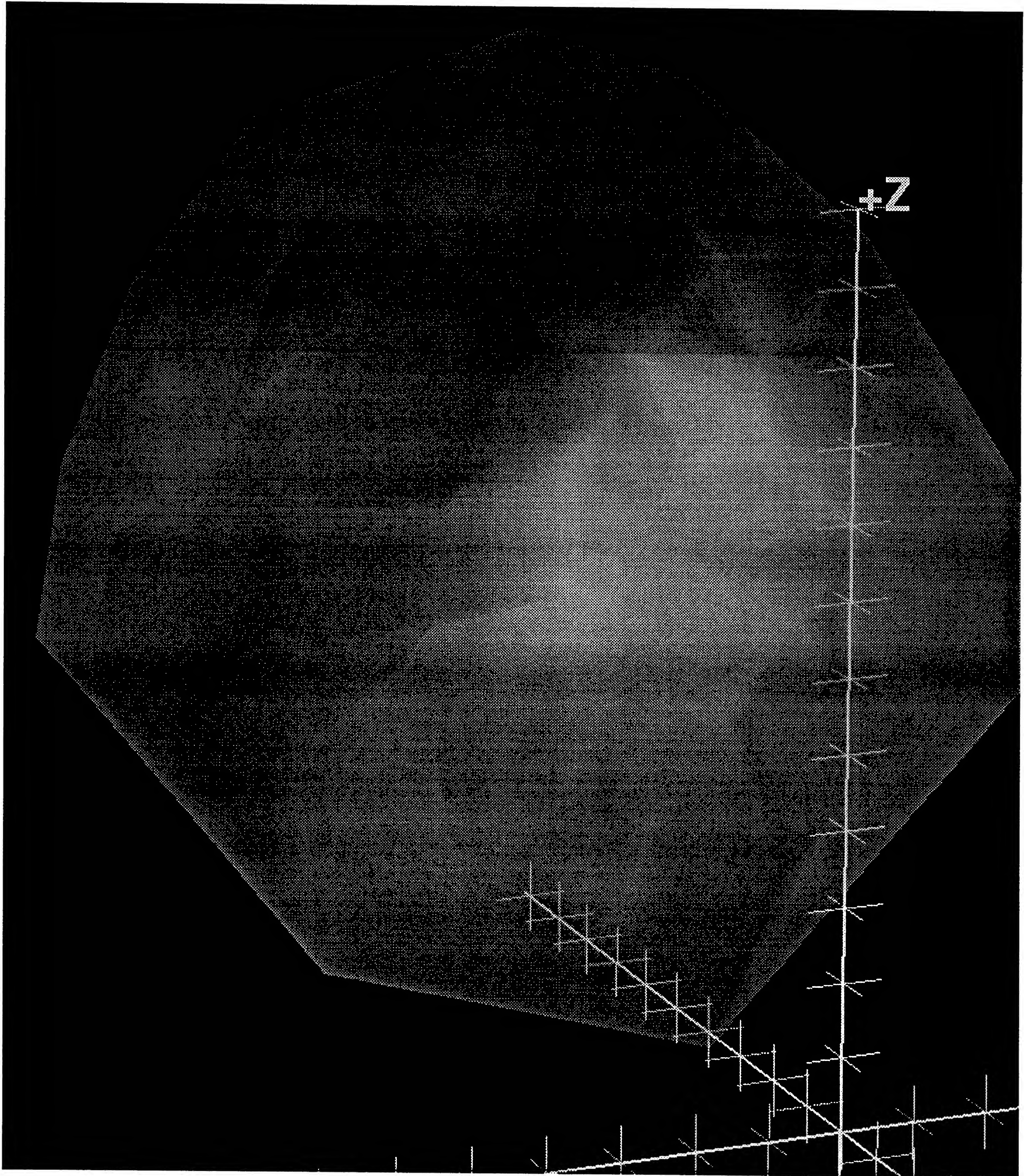


Figure 8.7: View from behind the head, looking toward the nose of the magnetic field pattern over the auditory cortex of the head.



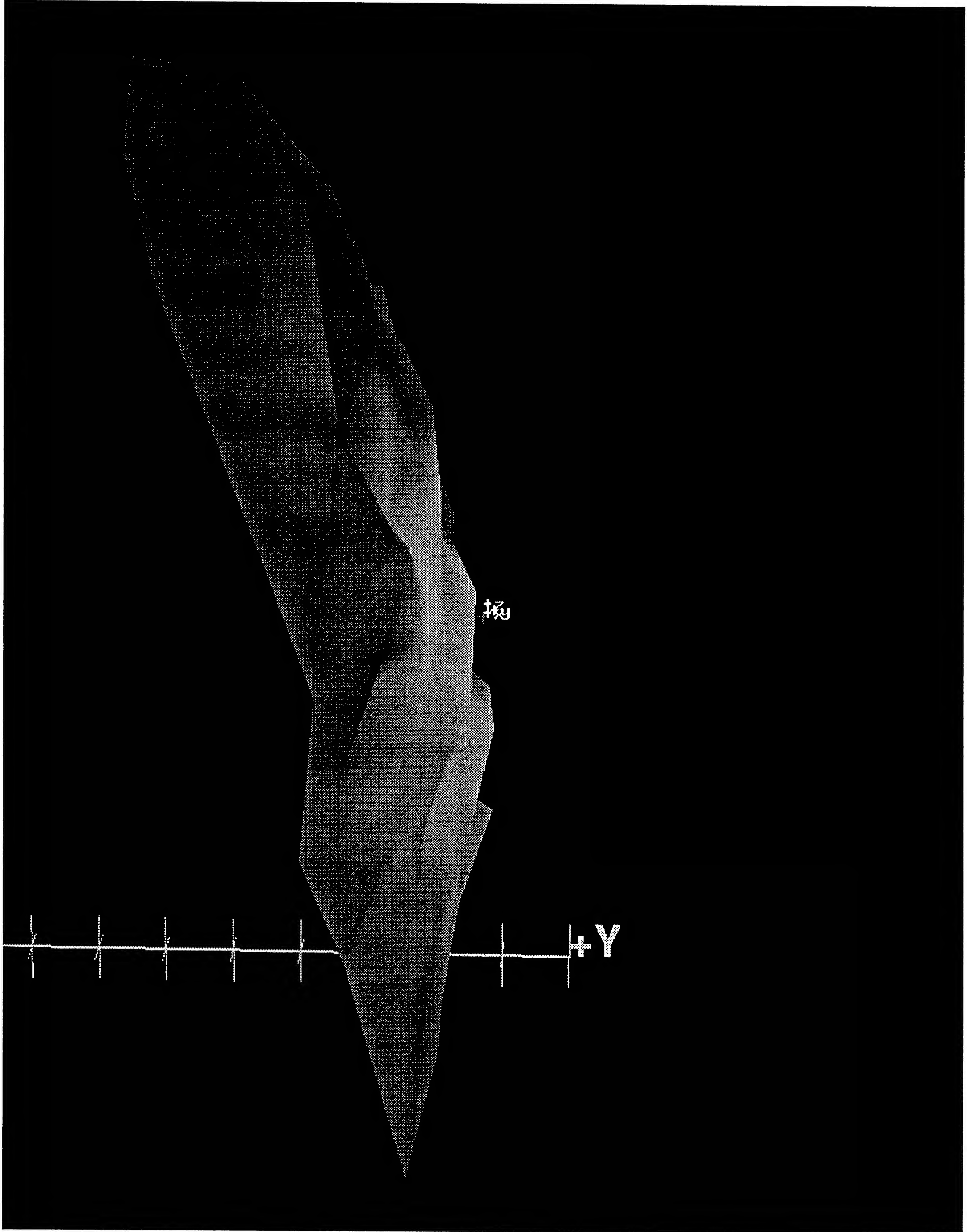


Figure 8.8: Lateral view of the magnetic field pattern from auditory cortex.



### 8.3 Forward Simulation Studies: Superficial Magnetic Fields Modeled from Deep Electric Currents

A simulation study was undertaken to provide a foundation for testing inverse theories with simple planar geometry constructs. The basic theory is well known for calculating the forward solution for a magnetic field generated from an electric current [247][248]. The generalized reverse problem, finding the volume electric currents from a superficially measured magnetic field, is not known. It has been classically been considered an under-determined problem from the days of Helmholtz's 1853 paper on the lack of a unique solution to the electrostatic inverse problem [249]. A number of numerical electric and magnetic field modeling algorithms are given in Steele [250]. We used the semi-infinite plane approach to starting our analysis of surface to surface field modeling. We show from the electric current in one surface, we can calculate the magnetic fields at another surface. We tested our algorithm and code on planar triangular meshes.

Figure 8.9 shows an example simulation. Two plane surfaces were constructed from uniform volume gradients. One plane was oriented vertically, from  $-z$  to 0, and the other was oriented horizontally, from  $-y$  to  $+y$ . The vertical plane simulates a sulcus wall plunging deep inside the brain. The horizontal plane represents the measurement surface of the squids above the scalp.

The color of the triangles represents the intensity and sign of the magnetic field. Green represents a negative magnetic field, Red represents a positive magnetic flux. The depth of the color represents the intensity of the flux. The flux isocontours are outlined at 100 fT intervals.



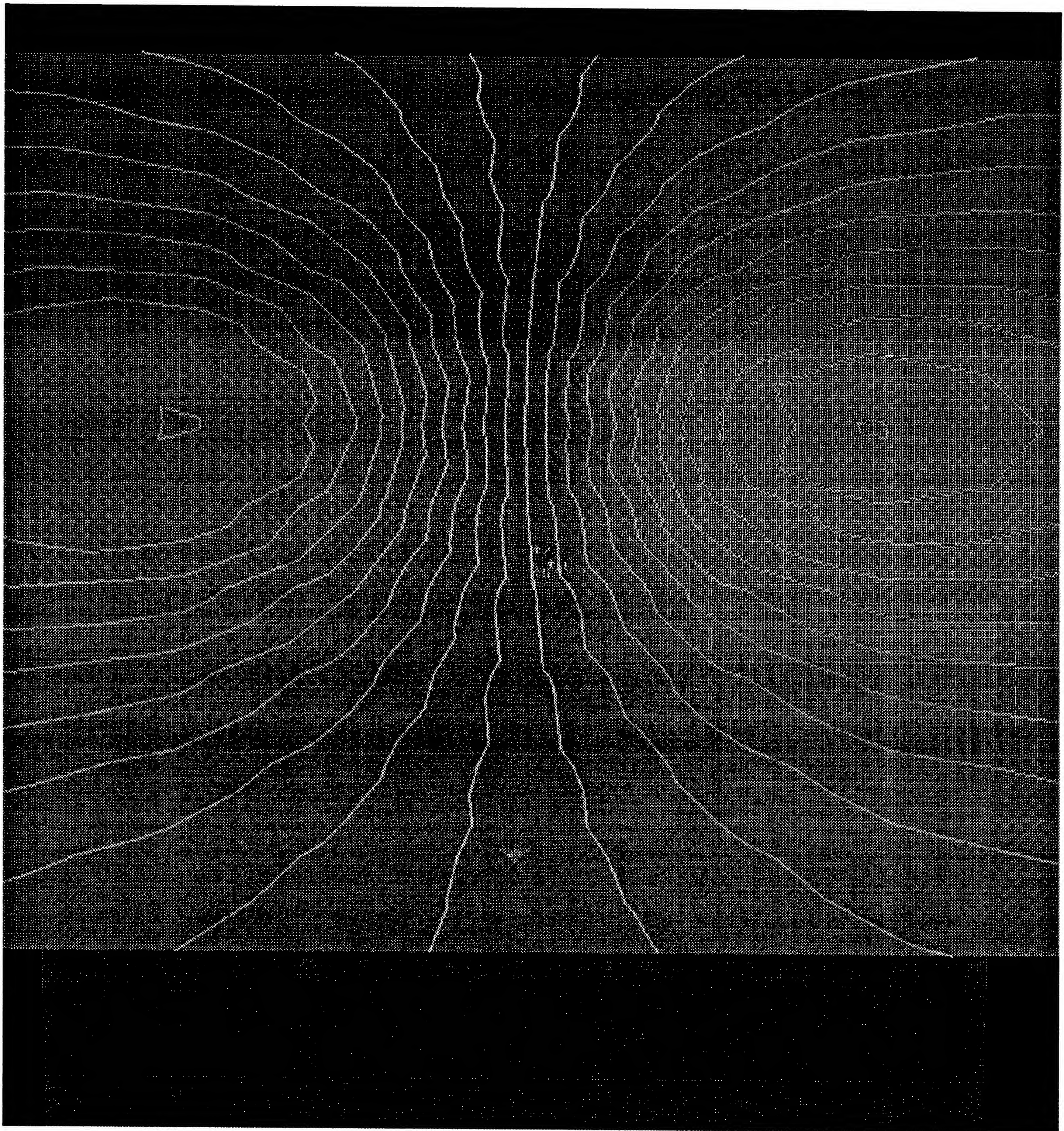


Figure 8.9: A simulation of the surface magnetic field.  
Simulation of the superficial magnetic field produced by a single electric dipole in a location that approximates the location of a deep brain electric source.



## 8.4 Comprehensive Source Searching: An Inverse Simulation Study

A novel inverse algorithm developed by Lu [251] called the Comprehensive Source Searching (CSS) was implemented in the SpiderWeb software package for estimating and visualizing the distribution of neuronal activity across the human cerebral cortex from measurements of the magnetic field pattern near the scalp. The procedure is based on two concepts. One is division of the image surface into “domains” whose size is defined by the spatial resolution of the given measurement set at each cortical region. The second is a probabilistic characterization of the magnetic inverse solution by a chi-square probability distribution. The algorithm interactively computes the probability distributions for one, two, or a small number of separate sources. These sources are constrained to lie on the cortical surface, which is constructed from MRI measurements using an algorithm such as the ‘SpiderWeb’ algorithm. All the results from the inverse solution can be visualized and animated with a graphical user interface in the software package.

### 8.4.1 Introduction

A magnetic source image (MSI) characterizes the distribution of neuronal electrical activity in the human brain from an analysis of non-invasive measurements of the magnetic field pattern over the scalp. Determining an MSI from measurements of a field pattern is called the magnetic inverse problem. Like several other types of inverse problems, a variety of image current distributions can explain the measurements. The non-uniqueness has been proven by Helmholtz 1853 [249].

While it is true there is no unique inverse solution to the generalized magnetic inverse problem, methods exist to at least partially overcome it. The most common way to ‘solve’ the inverse problem is to characterize the image current distribution with a small number of parameters. However, it is almost impossible to define a source model that is simultaneously restrictive enough to make the problem unique and also capable of describing the essential features of the current distribution. A widely used model, suitable for interpreting the simplest field patterns, is the equivalent current dipole, which is based on the assumption that the distribution of neuronal activity responsible for the measurements is confined within a small region (or within several well-separated regions) compared with the distance between the



center of activity and the sensor locations. While the equivalent current dipole model has been a great success in the early days of neuromagnetic research, and it is still producing useful results, misleading solutions can be obtained if the assumptions that sources are confined can not be justified.

Another way to overcome the non-uniqueness of the magnetic inverse problem is by introducing *a priori* information about the possible source configuration, and using some statistical criteria to eliminate “magnetically silent sources”, such as sources whose current is oriented perpendicular to the scalp. Hamalainen and Ilmoniemi (1984 and 1991) [252] [253] have introduced a general approach for the planar geometries that exploits lead-field analysis in a linear estimation based on the image current distribution having the minimum norm. The minimum norm criterion selects the current density whose mean square strength integrated over all space has a minimum value. This criterion eliminates the ambiguity caused by magnetically silent sources. In practice, however, minimum norm estimates may be unstable with respect to modeling errors and measurement noise. Moreover, with this procedure no unique depth can be determined for planar surface where the image current distribution is represented.

The question is to what extent we can obtain *a priori* information about the source geometry. From physiology, we know that the predominant source of extracranial magnetic fields is the intracellular current whose flow is confined by cell membranes. Evidence for this comes from relative polarities of sensory evoked potentials and magnetic fields (Hari *et al.*, 1980) [254] and studies of isolated nerve preparations (Wikswo *et al.*, 1980; Plonsey, 1981; Roth and Wikswo, 1985) [255][256][257]. We also know that many areas of the cortex are organized in layers characterized by the types of neurons they contain and into columnar units with their axes transverse to the cortical layers. While many types of neurons take part in the activities of cerebral cortex, it is believed (but not proven) that only pyramidal cells contribute to extracranial neuromagnetic fields. Not only is this the largest cell population in cerebral cortex but the cells have a preferred orientation, with axes generally aligned perpendicular to the cortical surface. So, from physiological considerations, a simple picture of the generators of neuromagnetic fields emerges. The primary source of the extracranial magnetic field is the intracellular current distribution of an ensemble of pyramidal cells that is oriented perpendicular to the surface of the local cortical surface. They can be modeled as an ensemble of current elements pointing normal to the local surface.

A “parcelization” procedure divides the cortical surface into surface elements called “domains” whose size is defined by the spatial resolution for locating a

current dipole source placed at the element, taking into account the geometry of the magnetic sensor array and measurement noise level. This procedure greatly reduces the enormous number of current elements that need to be considered. For instance, if the number were to be based on the size in a direct 3-D reconstruction of cortical surface from MRI recordings with 1 mm resolution, a cortical surface area of  $4\text{ cm} \times 4\text{ cm}$  will result in about 10,000 triangles from the ‘SpiderWeb’ algorithm. Using an MRI with coarser resolution as the input to the 3-dimensional reconstruction program will result in a distorted cortical surface which most of the procedures for obtaining an inverse solution can not tolerate. With the “parcelization” procedure, the fine resolution of the cortical surface can be preserved and the number of current elements needed to be considered in the inverse solution becomes reasonable. In this sense, it is a general procedure that can be combined with other approaches, such as the MNLS procedure developed by Wang *et al.* (1992). Simulation studies with 72 sensors in a square array with spacing of 1 cm show that the size of a quantal at a depth of 3 cm is about  $1\text{ cm} \times 1\text{ cm}$ , while at a depth of 5 cm it is about  $2\text{ cm} \times 2\text{ cm}$ .

The CSS procedure is a novel statistical approach to the magnetic inverse problem. The algorithm interactively searches through the image surface consisting of all the possible current elements after parcelization of the cortical surface into domains whose size is defined by the spatial resolution of the measurement. It colors each domain of the image surface according to the highest  $\chi^2$  probability a source lying on that domain can offer. The resulted image of the  $\chi^2$  probability distribution shows not only the centers of neuronal activity, but also the distribution of uncertainties in the solution. The algorithm has been shown to be highly robust with respect to noise level in the measurements of magnetic fields and errors in determining the cortical surface.

## 8.4.2 Method

### 8.4.3 Parcelization of Image Surface

The spatial resolution for a given neuromagnetic measurement is limited by the noise level in the recording and the configuration of the geometry of the sensor array. If we define the image surface as the surface where we expect to find the sources that are responsible for the extracranial magnetic field pattern, the size of the domain at a given spatial location on this image surface should not be smaller than the spatial resolution of the given measurement at that location. The spatial resolution



for a measurement changes from one location to another. Generally speaking, the resolution will become coarser when the image location is further away from the measurement surface.

While there could be many ways to define the spatial resolution of a given measurement and divide the image surface according to it, the criterion selected here is: for a given root mean square signal-to-noise ratio  $R_{sn}$ , two elements  $i$  and  $j$  on the image surface belong to a same domain if the magnetic fields produced by them at all the measurement locations have a correlation  $c_{ij}$  that satisfies:

$$(8.2) \quad c_{ij} > c_{cr}$$

with

$$(8.3) \quad c_{cr} = 1 - 1/R_{sn}^2$$

and  $c_{ij}$  defined as:

$$(8.4) \quad c_{ij} = \frac{\sum_{k=1}^N B(i, k)B(j, k)}{\sqrt{\sum_{k=1}^N B(i, k)^2 \sum_{k=1}^N B(j, k)^2}}$$

$B(i, k)$  denotes the magnetic field at measurement location  $k$  generated by the image surface element at location  $i$ . The summation is performed over all the  $N$  measurement locations. The critical correlation  $c_{cr}$  is the correlation between two measurements of the magnetic fields generated by the same source current distribution under the influence of measurement noise.

A domain is thus defined as the ensemble of the image surface elements with pair-wise correlation exceeding the critical correlation. The parcelled image surface consists of domains each with one possible magnetic source residing at the geometrical center of it. The parcelization starts from the shallowest locations on the image surface, then sweeps through the whole image surface region where sources for measured magnetic fields are expected.

#### 8.4.3.1 Comprehensive Source Searching Algorithm

For  $N$  measurement positions and  $M$  sources, the magnetic field at the measurement position  $i$  can be expressed as

$$(8.5) \quad B_i^t = \sum_{j=1}^M A_{ij}Q_j$$



While  $A_{ij}$  is a purely geometry determined number which is defined by the relative position, relative orientation and the boundary between the current dipole and the magnetic sensor, and  $Q_j$  is the strength of current dipole in domain  $j$ . For different head models and different sensor geometries, such as the type of the gradiometer of the detection coil of the sensor,  $A_{ij}$  has different expressions.

In matrix format:

$$(8.6) \quad \begin{pmatrix} B_1^t \\ B_2^t \\ \vdots \\ B_N^t \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1M} \\ A_{21} & A_{22} & \cdots & A_{2M} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NM} \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_M \end{pmatrix}$$

The only unknowns in this problem are the elements  $Q_j (j = 1, M)$ . If we define the best solution as the source configuration that has the minimum  $\chi^2$  with respect to the theoretical and measured magnetic fields, the equation to be satisfied is:

$$(8.7) \quad \frac{\partial \chi^2}{\partial Q_j} = 0$$

Given the definition of  $\chi^2$  as:

$$(8.8) \quad \chi^2 = \sum_{i=1}^N \frac{(B_i^t - B_i^m)^2}{\sigma_i^2}$$

where  $B_i^t$ ,  $B_i^m$  and  $\sigma_i$  are calculated and measured magnetic field and standard deviation of the averaged measured field. The following linear equation can be obtained:

$$(8.9) \quad C \cdot Q = D$$

where

$$(8.10) \quad C_{ij} = \sum_{k=1}^N A_{ki} \cdot A_{kj} / \sigma_i^2$$

$$(8.11) \quad D_i = \sum_{k=1}^N A_{ki} B_i^m / \sigma_i^2$$

The basic idea of the ‘‘Comprehensive Source Searching’’ technique is that any kind of current source distribution can be modeled as a distribution of a minimum number of current elements, and statistical tests can provide a fairly good means to select

the best suitable solution to a given inverse problem. By assuming that there is one candidate for the image solution at each domain, the program searches from possible combinations of candidates for the best estimation of the inverse solution. The algorithm interactively computes the  $\chi^2$  probability distributions for one, two, or a small number of distinct domains. It colors a display of the image surface of each domain according to the best  $\chi^2$  probability that a source lying in that domain can offer. The resulted image of the  $\chi^2$  probability distribution shows not only the centers of neuronal activity, but also the distribution of uncertainties in the solution. The coloring of cortical surface provides a pictorial representation of the magnetic source image. It can also define equal probability contours as a new way to characterize an inverse solution.

#### 8.4.4 Image Surface Parcelization

Assuming that the root mean signal-to-noise ratio  $R_{sn}$  is 10, as in most of the neuromagnetic measurements, the result from the image surface parcelization procedure is illustrated in Figure 3.2. The upper edge of the vertical image surface is only 2 mm below the measurement surface. The result of the parcelization is shown as parcelization of the image surface.

The size of a parcel increases with depth below the measurement surface. When it is only 2 mm deep, the size of the quantal is less than  $10 \text{ mm}^2$ . At a depth of 3 to 4 cm, which is the depth for most of the neuromagnetic studies on human cerebral cortex, the size of a domain is about  $1 \text{ cm}^2$ . The spatial resolution is about  $2 \text{ cm}^2$  at a very large depth.

With the parcelization procedure, the number of image elements needed to be considered in the inverse solution step is greatly reduced. In our case, we start with 540 image elements, and find out the resulting number of domains is only 130. 70 % of them are distributed in the region with depth less than 2 cm, which can be excluded in studies of activity in human cerebral cortex because it is occupied by the vacuum section of the dewar, the scalp and the skull. If we restrict ourselves only to the region of depth from about 2 cm to 5 cm, the number of domains is about 30. That can be conveniently handled by many inverse solution procedures.

### 8.4.5 An Example of Applying CSS

The comprehensive source searching procedure is carried out after the image surface is parcelled. The program iteratively puts one source on the center of each domain in turn and scales the source strength to achieve the minimum chi-square between the calculated magnetic field and the measured. It returns the chi-square probability for each domain and re-colors the image surface according to the probabilities. The result is shown in Figure 8.10, where the probability that a source lies in the central outlined region exceeds 0.2. In the Figure, the horizontal surface is the measurement surface. The location of a single current dipole source is indicated by the triangle on the vertical plane, its orientation is the same as the surface norm of the triangle, and it has a depth of 3 cm from the horizontal plane. The measurement surface has a size of 6 cm×12 cm with sampling space of 1 cm. The image surface has a dimension of 6 cm×9 cm populated with identical triangle of size 10 mm<sup>2</sup>. The width of the region of probability exceeding 0.2 is about 2 cm. The width of the region with probability above 0.9 has a width of 1 cm with the actual source lying at the center of it.

## 8.5 Discussion

The quantization procedure can be viewed as a pre-processing procedure for almost all the procedures that solve magnetic inverse problem. It can greatly reduce the amount of computation in inverse solution calculations without loss of resolution. For example, it can be used as a first step toward using MNLS algorithm. Pre-processing the image surface with the parcelization procedure before the MNLS procedure not only reduces the dimension of the matrix one needs to find the inverse matrix for, it also assigns different weights for sources at different depths. This is because the domains have different sizes at different

The critical correlation  $c_{cr}$  introduced in the parcelization procedure can be generalized to serve as an objective criterion for the quality of dipole fitting procedures conventionally used in neuromagnetic studies. It is directly defined by the signal-to-noise ratio in a measurement, and has the virtue of simplicity and objectivity. The idea of parcelization can also be generalized as a tool to predict the sensor array configuration and signal-to-noise ratio needed to achieve a certain spatial resolution goal for a measurement. This has to be done for the specific measurement system being used.



The image resulting from the comprehensive source searching procedure can be further sharpened. A simple way is to use the  $n$ th power of the chi-square probability as the index for color. The reason is that the chi-square probability is always less than or equal to 1.0. Comprehensive source searching can also be used as a way to replace the Monte Carlo simulation to estimate uncertainties in source localizations. Comparing with the conventional dipole fitting procedures, it guarantees that the absolute minimum chi-square point in the solution space can be reached.

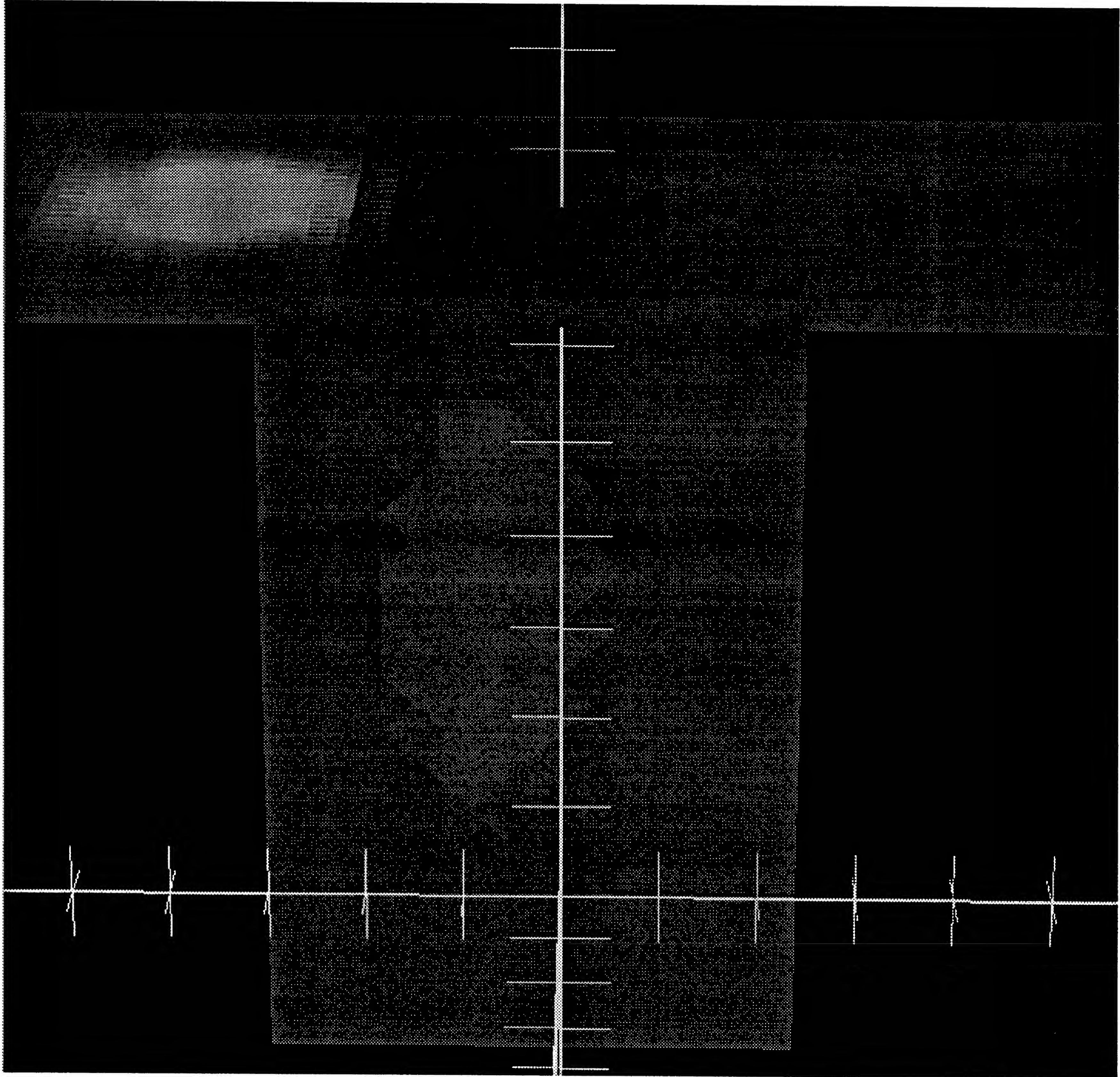


Figure 8.10: The Comprehensive Source Searching (CSS) inverse solution algorithm of Lü [251]. The bottom plane represents the estimated source configuration that would generate the given magnetic field on the upper, perpendicular plane.

## Chapter 9

# Alphon Location by Registering MRI and MSI Coordinate Systems

### 9.1 Introduction to Alphon Activity Localization

Alpha rhythm recorded magnetically over the parieto-occipital area of the human scalp is found to consist of a sequence of oscillatory bursts with the neuronal source of each burst lying near the longitudinal fissure. The temporal period for oscillations within a burst is remarkably stable, and is distinguishable from narrow-band filtered white noise. The field pattern for the individual oscillations that comprise a burst is also remarkably stable but differs significantly from one burst to the next. Magnetic source imaging identified the cortical locations of the center of activity for each oscillation and showed that it was stable to within 3 mm for the central 4 - 5 oscillations of a burst. These characteristics suggest that the origin of each burst is a stationary, spatially-coherent cortical excitation, which we call an alphon. Using covariance techniques to improve the estimate for the amplitude registered by each magnetic sensor, a current-dipole source model and also an extended-source model were fit to the field pattern of each alphon. Both models provide evidence that the coherence length of neuronal activity extends across cerebral cortex for 3 cm. This is specified by the extended-source model with a direct fit to the spatial pattern of the magnetic field data.



## 9.2 Introduction

The first electrical signal from the human brain recorded by Berger in 1929 [258] [259] was the spontaneous activity now known as the alpha rhythm (Berger, 1930). Despite the fact that it is often the dominant signal registered in the electroencephalogram (EEG) and magnetoencephalogram (MEG), comparatively little is known about the relative contributions from various functional areas of human cerebral cortex. The importance of understanding spontaneous rhythms comes in part because they arise from many regions of human cerebral cortex, and they express interactions with the thalamus (Andersen and Andersson 1968; [260] Steriade *et al.* 1990; [261] Lopes da Silva 1991 [262]). In addition to alpha rhythm of occipital and parietal cortices, similar phenomena are expressed as the mu rhythm of the central cortex by Chatrian 1959 [263] and the recently identified tau rhythm of auditory cortex (Tiihonen *et al.* 1991 [264] Kaufman *et al.* 1992[265]). Interest in spontaneous rhythms is also motivated by their correlation with normal and abnormal brain functions (*e.g.*, John *et al.* 1988 [266]). In addition, spontaneous activity is locally suppressed when a cortical region is involved in a sensory or cognitive task. Using EEG recordings of mu rhythm over the sensorimotor area, Pfurtscheller and Aranibar 1977 [267] observed this effect, which they called event-related desynchronization. Pfurtscheller *et al.* [268] also observed desynchronization of mu rhythm over the contralateral central area during planning of voluntary one-sided finger movement, beginning about 1-2s before movement onset. We prefer the term ‘suppression’ because it does not impose a presumption as to the precise mode of decay in EEG or MEG power.

A close relation between the location and timing of suppression of spontaneous rhythms with cognitive processes was first observed by Kaufman *et al.* 1990 [269]. In neuromagnetic studies suppression of alpha rhythm was observed over occipital and parietal areas of the scalp when a subject engaged in a mental imagery task. Of particular interest is the finding that suppression is enhanced near the midline of the occipital scalp, thus indicating that visual cortex is involved in the task. In subsequent studies, Kaufman *et al.* 1991 [270] found that the level of alpha band activity in the MEG originating in auditory cortex is sharply reduced when subjects scan memory for previously heard tones. Furthermore, this suppression exhibits a hemispheric asymmetry. While suppression occurs bilaterally, only the duration of suppression over the right hemisphere covaries significantly with the size of the memory set for all of their subjects. The fact that the covariation is poor for the left hemisphere provides evidence against a picture of global alpha blockage.

The cortical origin of alpha rhythm is well established through studies of

the transcortical voltage profile in the visual cortex of dog [271]. Evidence suggests that alpha rhythm is initiated in epicenters by thalamic projections [260] [262] [261] from which the cortical activity spreads in various directions by way of cortico-cortical connections. The measured spatial coherence length across visual cortex is about 6 mm in dog. In addition, Lopes da Silva and Van Leeuwen (1977) [272] demonstrated in electrode penetrations through cortex that the phase of alpha rhythm changes by  $180^\circ$  at a depth of about 1 mm below the cortical surface, indicating that the neuronal generators of alpha rhythm should be at the level of the large pyramidal cells. In a recent study to determine the origin of alpha rhythms observed in the EEG, Nunez 1989 [273] showed that it is necessary to assume a set of clustered dipoles, *i.e.*, dipoles with similar orientation that are activated in a coordinated way, to account for the observed pattern at the scalp. The sequential generation of such sets of clustered dipoles with a constant phase lag between sets can be accounted for, at least for the case of the alpha rhythm, by the model of a cortical chain of neurons interacted by means of cortico-cortical fiber systems [274]. In a comparative physiological study, Bullock and McClune (1989) [275] showed that on average for the cortex of rabbits and rats, coherence of the spontaneous EEG when no rhythmic activity is dominant tends to decrease to a value of 0.5 within a distance of 2.5-5 mm for most frequency bands. These authors stress, on the basis of these findings, that the EEG generators are largely independent but synchronized to varying degrees. It appears that in those cases where rhythmic activity dominates the EEG, as expressed by the alpha rhythm, the degree of cooperativity may increase in a very significant way, such that coherent activity can extend over larger regions of cortical surface. However, the question still remains about the nature of human alpha rhythms, such as the specific cortical locations that support this activity and the spatial configuration of rhythmic activity within a given cortical area.

Since magnetic source imaging provides a non-invasive technique for monitoring human brain activity with a high accuracy in source localization, [148] over the last decade, several attempts have been made to localize the sources of human alpha activity with magnetic measurements. These include measurements of the covariance between the EEG and a single-sensor MEG (Chapman *et al.* 1984); [276] observations of individual spindles with a 4-sensor system (Vvedensky *et al.* 1985) [277] where the period stability of successive oscillations within a burst was noticed; demonstration of significant differences in spatial characteristics among cortical sources of individual bursts (Ilmoniemi *et al.* 1988) [278]; and determination of the distribution of cortical strengths (measured as a current dipole moment) for individual spindles and source positions (Williamson *et al.* 1989) [279]. The latter two studies provide evidence that human parieto-occipital alpha rhythm consists of distinct cortical excitations, with sources of successive spindles having different geometries (position, orientation,



and/or spatial extent). To emphasize this point of view, the term “alphon” was introduced for such spatially-coherent cortical excitations [279].

Opposed to this view is Dick’s [280] notion that the time-series describing the sequence of spindles that comprises the alpha rhythm cannot be distinguished from narrow-band filtered gaussian noise. Dick found in an analysis of the EEG that the amplitude histogram is fairly well represented by a Rayleigh probability distribution. This distribution has but one parameter, so the amplitude and standard deviation of the distribution are related by a multiplicative constant. This feature was later verified for MEG measurements [281].

The distinction as to whether alpha rhythm can be considered “noise”, or a signal of low dimensionality, or is structured in a more specific way is of fundamental importance for the understanding of the behavior of the generator of alpha rhythm. Such distinctions are also essential for any kind of development of mathematical approaches to identify and classify different characteristics of alpha rhythm.

However, such a description in terms of narrow-band filtered noise only focus on a static picture of the statistical properties of the amplitudes, while overlooking the underlying dynamic processes as may be expressed by the appearance of a succession of spindles in the time series.

We report here that the neuronal source of individual oscillations within a well-developed alpha spindle can exhibit a remarkable spatial stability across cortex. Computer fits of the observed field patterns representing individual oscillations within a spindle with a current-dipole source model and with an extended-source model provide consistent estimates for the spatial extent across cerebral cortex of an alphon’s neuronal activity, thereby providing a gauge of its coherence length.

## 9.3 Method

Four male laboratory personnel (ages from 23 to 52) served as subjects. Most of the recordings were made with a pair of probes each containing seven dc-SQUID sensors (Model 607 neuromagnetometer, Biomagnetic Technologies, Inc.) positioned over the left and the right parietal or occipital areas of the scalp. The detection coils were second-order gradiometers with 1.5 cm diameter and 4 cm baseline, and the sensor noise level was about  $20 \text{ fT Hz}^{-1/2}$  for most channels, while one or two exhibited noise



as high as  $50 \text{ fT Hz}^{-1/2}$ . Individual sensors were calibrated with a relative accuracy of better than 1%. With the subject prone and alert, recordings were made within the bandwidth of 0.5-50 Hz for 16 s epochs of spontaneous activity with eyes closed. The total level of instrument and subject noise was determined with the subject viewing an interesting static display color graphics. Data were digitally filtered to pass the bandwidth 8-13 Hz, and a computer routine was used to identify segments of the time series where the RMS amplitude within the middle 12 s of the epoch significantly exceeds the noise level.

Positions and orientations of alphas is registered in the PPN Headframe [233] described in Section 5.4.2 of this dissertation. This method, together with data from the 14-sensor system can provide 3 mm accuracy in determining the location of a current dipole at a depth of about 4 cm in a head modeled as a conducting sphere (Yamamoto *et al.* 1988 [148]). This enables us to compare directly the deduced locations and spread of the dipole sources with the anatomical structures of an individual's cortex.

To ensure accuracy in locating each source of alpha rhythm, it was important to achieve a high signal-to-noise ratio, because only a small area of the field pattern is measured. By positioning the neuromagnetometer probes so that field extrema of the individual spindles are close by, the source positions in three dimensions can be determined without need to move the dewars. The accuracy of this "fixed position" technique when probes are placed directly over the extrema has been analyzed by Costa Ribeiro *et al.* [282] who considered the cases of a single probe with 5 or 7 sensors and two probes with 7 sensors each. However, this is the ideal situation, when determining the locations and field strengths of the two field extrema is sufficient to determine all 5 dipole parameters [283]. If an extremum is not within the areas of each probe, the accuracy in deducing the location of the neuronal source greatly diminishes.

The neuronal source of a spindle hereafter "alphon" was determined for spindles that satisfy the following criteria:

1. Field component normal to the scalp has opposite polarity for the two probes to ensure that both extrema are observed;
2. Peak RMS amplitude averaged across 14 sensors exceeds 400 fT to ensure relative high signal-to-noise ratio;

3. Mean RMS amplitude across sensors exceeds 150 fT to ensure that all the sensors are sufficiently close to the two field extrema;
4. Period between zero crossings<sup>1</sup> of the signal registered in each of the 14 sensors is stable to within 5% throughout the duration of the spindle (temporal stability);
5. Correlations between the 14-dimensional signal vector representing peak activity at one moment for all the 14 sensors and the 14-dimensional signal vectors of all the other successive peaks within the same spindle exceeds 0.9 (spatial stability).

The last criterion eliminates cases where the latter portion of a spindle is overlapped by the initial portion of the following spindle.

Once an alphon is identified, the best estimate for the amplitude registered by a given sensor is obtained from the temporal covariance between its signal and each of the signals from the other sensors[278]. It should be emphasized that the last criteria in the preceding list defining the signal from an alphon is extremely important. This justifies the covariance calculation within a spindle, which necessarily extends over a certain duration of time. The source of the measured field pattern was first characterized by the standard model for a current dipole, and the corresponding 5 parameters were determined with the head represented by a uniform conducting sphere whose center coincides with the center of curvature of the inner surface of the skull in the parietal-occipital area. In addition, an extensive-source model was used to estimate directly the active cortical area that produced the measured field. The simplest extensive-source model represents activity by a  $3 \times 3$  square array of 9 equally-spaced current dipoles of identical strength oriented perpendicular to a common plane. There are 6 parameters to be determined in this model 3 for the position of the center dipole, 1 for the tilt of the plane with respect to the local radius of the sphere that best fits the portion of the scalp where the measurements were taken, 1 for the strength of each dipole, and 1 for the spacing of the array. This model provided a significantly improved fit compared with that for a single-dipole model, as indicated by chi-square statistics.

The locations of the three cardinal landmarks can be established accurately by computer analysis of each Pointer's spots on the MR images to determine their points of convergence. The point of convergence can be determined within 1 mm in the MRI coordinate system, so the overall accuracy in location is limited by how accurately each pointer is held against its cardinal landmark and the resolution of the

---

<sup>1</sup>all subsequent use of "period" has the same meaning



MRI system. The overall accuracy for specifying alphon positions within the head is better than about 2 mm (See Section 5 of this Dissertation for more details). This enables us to compare directly the deduced locations and spread of the dipole sources with the anatomical structures of an individual's cortex.

## **9.4 Results**

With sensors positioned to record alpha rhythm from the parieto-occipital sulcus, the time-series was compared with that of narrow-band filtered gaussian noise. For this comparison, continuous time-series were used, without imposing conditions for minimum amplitude.

### **9.4.1 Comparison with Narrow-Band Filtered Noise**

After filtering the alpha and noise signals to retain only components with the 8-13 Hz bandwidth, the distribution of the amplitudes of the recordings are found to be well-described by a gaussian distribution. These plots give the cumulative distribution of amplitudes. This result implies that the amplitudes can be described by gaussian distributed random variables, in agreement with the conclusions of Dick and Vaughn 1970 [280].

### **9.4.2 Location Stability**

For subjects exhibiting strong alpha rhythm, bursts satisfying our strict criteria can be identified for about 20% of a recorded time series. The principal limitation on the percentage of bursts that satisfy the alphon criteria is the fact that our pair of probes covers an insufficient area of the scalp to always register two extrema of the field pattern within the array of sensors. For a much higher percentage of bursts an extremum is found under one probe but the other extremum is near the edge or is just outside of the other probes's sensor array. This remained true when the probes were placed at various asymmetrical positions over left and right hemispheres with one farther from the inion or midline than the other.



Once an alphon is identified, the best estimate for the field amplitude registered by a given sensor is obtained by computing the temporal covariance between its signal and each of the signals from the other sensors. The uncertainty in position (95% confidence) is typically about 0.8 cm in radial position and 1.3 cm in distance along  $X$  and  $Z$  axes. The uncertainties are imposed by the relatively low signal-to-noise ratio for these unaveraged data. Most alphons lie within 5 cm of the midsagittal plane.

#### 9.4.2.1 Discussion of Tables and Figures

Table 9.1 shows the distribution of Alphons recorded for this study. Table 9.2 gives the values for Strong Alphons, where the signal was  $> 80$  fT. Left hemisphere alphons have  $+Y$  values and Right hemisphere alphons have  $-Y$  values in the PPN Headframe. Table 9.3 lists medium strength ( $80 < \text{Signal} < 40$ ), and Table 9.4 list the remaining weak  $< 40$  alphons. Figure 9.1 shows a MRI slice 3 mm to the right of the midsagittal plane, at about  $y = 0.3$ . Projected onto this plane are the alphons with  $y$  values  $> 0.0$ . These alphons were grouped into three classes, determined by their perpendicular distance to the nearest sulcus, as shown in Table 9.4.2.1. Figure 9.2 shows the slice 3 mm left of the  $y = 0$  plane. Table 9.4.2.1 gives three groups of these alphons with respect to the sulcus. Figure 9.3 illustrates a three dimensional reconstruction of the SpiderWeb surface in the vicinity of a strong Alphons about the parieto-occipital sulcus.

| Condition            | 1%   | 3%   | 5%   | 10%  | 30%  | 50%  |
|----------------------|------|------|------|------|------|------|
| Noise recording      | 15   | 447  | 1337 | 3966 | 7839 | 8316 |
| Alpha recording (SW) | 1100 | 2246 | 4794 | 6063 | 7999 | 8221 |
| Alpha recording (JH) | 2828 | 5294 | 6125 | 6904 | 7763 | 8109 |
| Alpha recording (ZL) | 2136 | 3891 | 4506 | 5228 | 6384 | 8171 |

Table 9.1: Number of periods that are stable to within the indicated percentage for gaussian noise within the bandwidth of 8 - 13 Hz and for alpha rhythm within the same bandwidth for each of 3 subjects. For both conditions (noise and alpha recordings), the total number of periods considered is 8318.

| X     | Y (left) | Z    | Signal (fT) |
|-------|----------|------|-------------|
| -5.20 | .26      | 2.68 | 116.79      |
| -3.89 | 1.42     | 4.30 | 89.49       |
| -2.00 | .36      | 8.44 | 132.45      |
| -1.33 | .52      | 9.13 | 139.20      |
| -1.92 | .97      | 9.31 | 110.84      |
| -1.64 | .54      | 7.20 | 214.68      |
| -4.81 | .45      | 6.69 | 86.02       |

| X     | Y (right) | Z    | Signal (fT) |
|-------|-----------|------|-------------|
| -3.82 | -.04      | 5.89 | 85.99       |
| -2.67 | -.08      | 7.76 | 81.07       |
| -2.84 | -.17      | 2.46 | 305.85      |
| -2.84 | -.25      | 7.66 | 112.15      |
| -3.12 | -.21      | 6.91 | 121.06      |
| -3.46 | -.44      | 7.02 | 86.87       |
| -3.04 | -.48      | 6.49 | 105.19      |
| -3.52 | -.53      | 6.79 | 81.87       |
| -3.50 | -.52      | 6.70 | 86.66       |
| -2.26 | -.66      | 5.54 | 207.73      |
| -2.58 | -.98      | 6.45 | 123.54      |
| -3.08 | -.85      | 7.03 | 101.11      |
| -2.60 | -.96      | 6.34 | 121.78      |
| -2.43 | -1.36     | 6.97 | 118.13      |
| -3.07 | -1.42     | 6.76 | 84.71       |

Table 9.2: Subject SW, A class (strong) Alphas, total  $n = 22$ . Left hemisphere  $n = 7$ . Right hemisphere  $n = 15$ .

| X     | Y (left) | Z    | Signal (fT) |
|-------|----------|------|-------------|
| -3.63 | .00      | 7.42 | 49.41       |
| -4.71 | .02      | 6.39 | 57.96       |
| -4.56 | .06      | 6.91 | 43.91       |
| -4.43 | .19      | 6.85 | 50.44       |
| -4.84 | .20      | 7.05 | 52.03       |
| -4.96 | .23      | 6.84 | 44.24       |
| -3.98 | .40      | 8.29 | 54.21       |
| -3.95 | .60      | 9.76 | 63.60       |
| -2.73 | .66      | 7.93 | 62.12       |

| X     | Y (right) | Z    | Signal (fT) |
|-------|-----------|------|-------------|
| -5.08 | -.10      | 5.88 | 59.21       |
| -5.23 | -.58      | 7.25 | 41.42       |
| -4.30 | -.41      | 7.25 | 46.05       |
| -5.12 | -.74      | 6.32 | 47.72       |
| -5.43 | -.22      | 6.48 | 48.07       |
| -4.80 | -.15      | 6.20 | 52.83       |
| -4.79 | -.81      | 6.58 | 47.21       |
| -5.31 | -.11      | 6.34 | 53.18       |
| -5.08 | -.16      | 6.64 | 53.70       |
| -5.29 | -.18      | 6.48 | 54.53       |
| -5.77 | -.63      | 6.46 | 45.88       |
| -5.31 | -.60      | 6.41 | 55.21       |
| -5.00 | -.14      | 6.11 | 61.59       |
| -2.98 | -1.47     | 7.56 | 75.58       |
| -4.92 | -.64      | 6.27 | 49.45       |
| -4.32 | -.12      | 8.04 | 58.98       |
| -3.61 | -.03      | 7.98 | 72.25       |
| -4.56 | -.46      | 6.06 | 70.53       |
| -3.50 | -.52      | 6.70 | 86.66       |
| -4.13 | -.40      | 7.93 | 46.70       |
| -5.10 | -.08      | 7.18 | 49.28       |
| -3.78 | -1.43     | 6.34 | 76.57       |
| -3.10 | -1.24     | 7.73 | 63.42       |
| -3.89 | -.99      | 4.94 | 62.50       |
| -3.66 | -.80      | 7.19 | 58.41       |
| -3.12 | -.63      | 6.98 | 73.41       |
| -3.46 | -.44      | 7.02 | 86.87       |
| -4.51 | -.30      | 7.85 | 46.52       |
| -4.01 | -.07      | 8.59 | 53.36       |
| -3.56 | -.59      | 6.33 | 60.10       |
| -4.33 | -.08      | 6.57 | 49.86       |
| -5.50 | -1.36     | 4.95 | 43.26       |
| -5.61 | -.51      | 4.09 | 54.55       |
| -4.97 | -.04      | 7.14 | 51.59       |
| -5.61 | -1.57     | 4.02 | 43.19       |

Table 9.3: Subject SW B class (medium) Alphas. Total number  $n=44$ . Left hemisphere alphas  $n = 7$ . Right hemisphere alphas  $n = 15$ .



| X     | Y (left) | Z    | Signal (fT) |
|-------|----------|------|-------------|
| -5.46 | .74      | 6.26 | 23.13       |
| -5.10 | .59      | 5.92 | 38.47       |
| -6.22 | .52      | 6.98 | 23.90       |
| -6.16 | .34      | 4.72 | 30.21       |
| -5.74 | .47      | 7.09 | 31.50       |
| -6.73 | .43      | 7.88 | 12.48       |
| -5.64 | .16      | 6.98 | 22.68       |
| -4.77 | .23      | 7.11 | 39.84       |
| -5.38 | .06      | 6.40 | 31.35       |

| X     | Y (right) | Z    | Signal (fT) |
|-------|-----------|------|-------------|
| -5.19 | -.08      | 6.99 | 34.19       |
| -5.88 | -.10      | 7.04 | 32.01       |
| -5.64 | -.21      | 6.36 | 37.62       |
| -4.58 | -.23      | 7.87 | 34.66       |
| -4.98 | -.25      | 7.92 | 37.14       |
| -7.18 | -.28      | 4.84 | 32.60       |
| -5.62 | -.28      | 7.25 | 31.15       |
| -5.51 | -.32      | 6.18 | 37.50       |
| -5.25 | -.32      | 7.60 | 25.16       |
| -4.55 | -.38      | 9.12 | 33.97       |
| -4.72 | -.40      | 6.73 | 37.79       |
| -5.21 | -.41      | 7.12 | 38.75       |
| -5.46 | -.45      | 7.07 | 39.46       |
| -5.68 | -.45      | 7.02 | 33.50       |
| -5.82 | -.45      | 5.89 | 34.60       |
| -4.83 | -.58      | 7.58 | 33.62       |
| -6.33 | -.60      | 6.06 | 22.86       |
| -5.03 | -.60      | 7.56 | 25.25       |
| -5.80 | -.63      | 6.68 | 27.77       |
| -7.36 | -.70      | 6.08 | 15.54       |
| -6.56 | -.76      | 6.29 | 31.96       |
| -5.97 | -.83      | 8.28 | 22.81       |
| -5.18 | -.82      | 7.41 | 35.85       |
| -4.94 | -.94      | 8.21 | 27.06       |
| -5.44 | -1.07     | 4.82 | 38.78       |
| -6.82 | -1.80     | 4.08 | 21.11       |
| -6.76 | -2.48     | 4.13 | 17.04       |
| -7.26 | -2.48     | 4.05 | 14.41       |
| -4.40 | -2.48     | 7.03 | 33.99       |

Table 9.4: Subject SW C class(weak) Alphons,  $n = 39$ . Left hemisphere  $n = 9$ . Right hemisphere  $n = 29$ .

| Alphon | Class | Distance Class      | $n$ |
|--------|-------|---------------------|-----|
| Yellow | A     | $0 \leq 3\text{mm}$ | 7   |
| Yellow | A     | $3 \leq 6\text{mm}$ | 8   |
| Yellow | A     | $> 6\text{mm}$      | 7   |
| Yellow | A     | <i>Total</i>        | 22  |
| Red    | B     | $0 \leq 3\text{mm}$ | 19  |
| Red    | B     | $3 \leq 6\text{mm}$ | 13  |
| Red    | B     | $> 6\text{mm}$      | 3   |
| Red    | B     | <i>Total</i>        | 35  |
| Brown  | C     | $0 \leq 3\text{mm}$ | 16  |
| Brown  | C     | $3 \leq 6\text{mm}$ | 8   |
| Brown  | C     | $> 6\text{mm}$      | 5   |
| Brown  | C     | <i>Total</i>        | 29  |

Table 9.5: Right hemisphere alphon ranked by perpendicular distance from nearest sulcus. Subject SW.



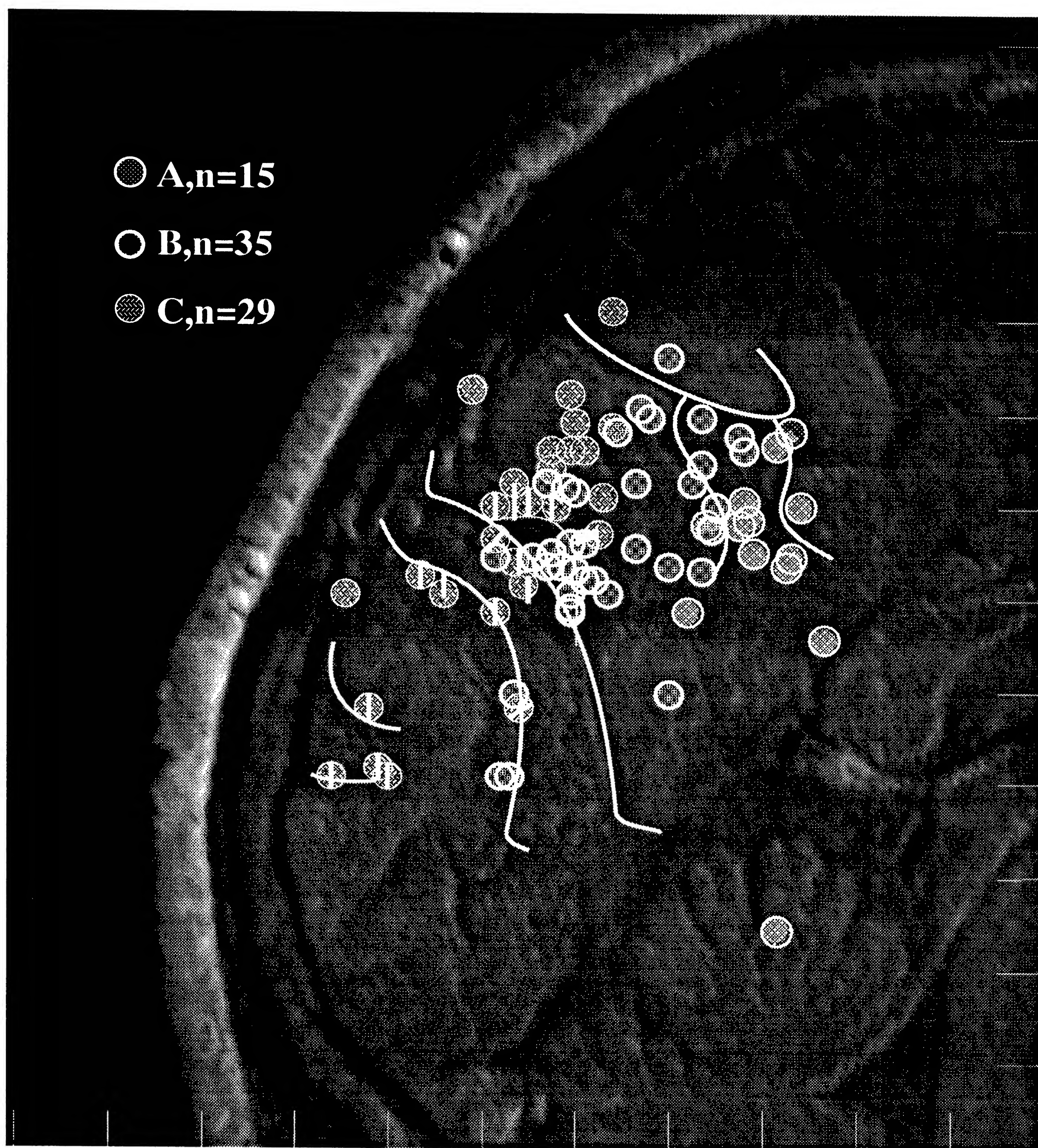


Figure 9.1: Subject SW, Right Hemisphere Alphons, viewed over a slice approximately 3 mm Right of the midsagittal plane.



| Alphon | Class | Distance Class      | $n$ |
|--------|-------|---------------------|-----|
| Yellow | A     | $0 \leq 3\text{mm}$ | 3   |
| Yellow | A     | $3 \leq 6\text{mm}$ | 3   |
| Yellow | A     | $> 6\text{mm}$      | 1   |
| Yellow | A     | <i>Total</i>        | 7   |
| Red    | B     | $0 \leq 3\text{mm}$ | 2   |
| Red    | B     | $3 \leq 6\text{mm}$ | 3   |
| Red    | B     | $> 6\text{mm}$      | 4   |
| Red    | B     | <i>Total</i>        | 9   |
| Brown  | C     | $0 \leq 3\text{mm}$ | 6   |
| Brown  | C     | $3 \leq 6\text{mm}$ | 3   |
| Brown  | C     | $> 6\text{mm}$      | 0   |
| Brown  | C     | <i>Total</i>        | 9   |

Table 9.6: Left alphons ranked by perpendicular distance from nearest sulcus. Subject SW.



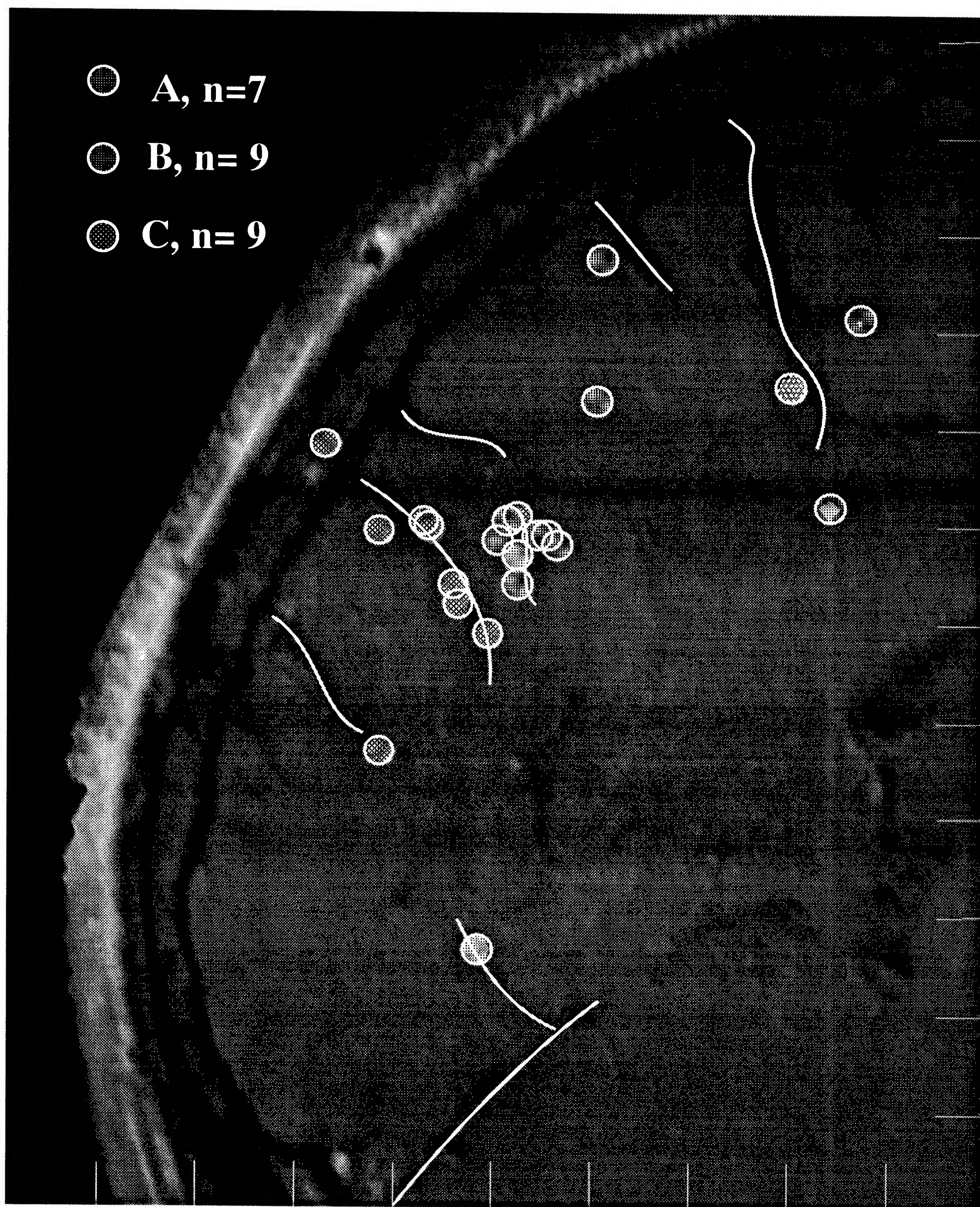


Figure 9.2: Subject SW, left hemisphere alphons.



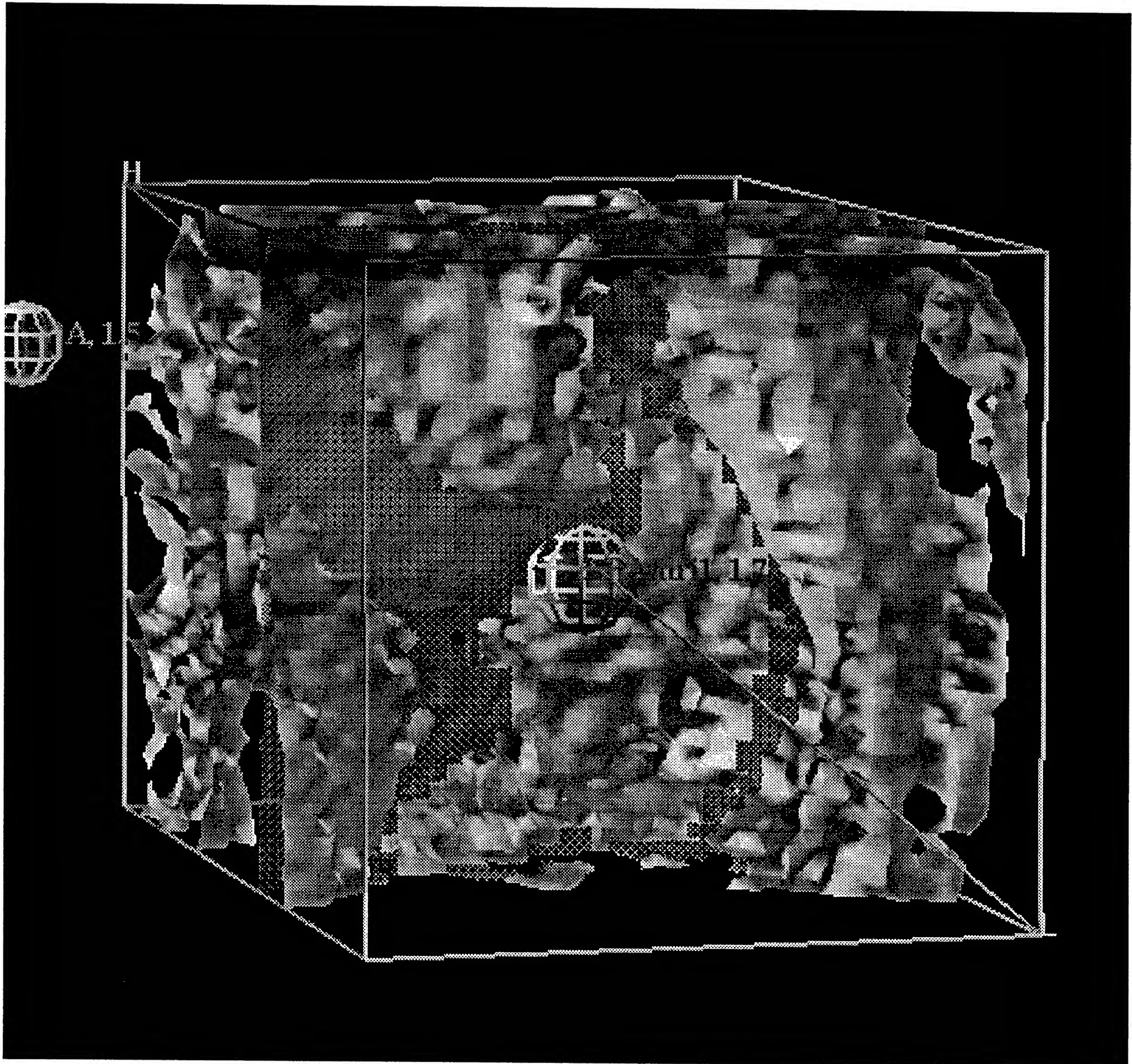


Figure 9.3: SpiderWeb surface about the parieto–occipital sulcus region of the alphon. The stippled planar surface is the reference slice about which the surface was grown.



## 9.5 Discussion

In this study, we observed that the lifetime of an alphon could be as long as 0.6 s. During its lifetime, the center of activity giving rise to successive oscillations is found to shift by less than 3 mm. This remarkable spatial stability, together with estimated activation area of several square centimeters, suggests that a spindle is not a happenstance arising from constructive interference of field patterns of many separately oscillating regions of cortex. Rather, it represents a coherent oscillation of a wide area of cortex. These moments of source stability may well correspond to the notion of “brain state” used by Lehmann (1987) to describe spatial configurations of alpha power across 19 EEG electrodes on the human scalp that remain stable for about 300 ms. His data show abrupt shifts in the global power pattern (and therefore the corresponding sources) from one brain state to the next. This high spatial stability could also serve as location “fingerprint” of alpha activity.

The spatial correlation length of 1-2 cm implied from our extended-source model for human sensory association areas may be compared with the coherence length of about 6 mm reported by Lopes da Silva and van Leeuwen (1978) from electrophysiological studies on dog visual cortex. The mediator of coherence may well be laminar-parallel intracortical fibers which extend 6-8 mm in dog (Lopes da Silva *et al.*, 1980) together with pyramidal collateral spread modules which extend over about 3 mm (Creutzfeldt 1978). Much is made in the literature of possible phenomenon of “synchronization” of alpha activity, whereby a subset of presumed independent cortical oscillators (Elul, 1972) may be brought into synchrony and thus produce a strong EEG because of constructive interference of their individual volume currents. Our estimate of the extent of coherent neuronal activity emphasizes that coherence is not just a chance occurrence of a small fraction of cortical neurons but entails intracortical interactions that produce rhythmic activity having long-range coherence.

The implication of this property is important when we consider the distribution of deduced alphon locations across cortex. Many alphas have their centers of activity located within 5 mm of the longitudinal sulcus, as determined from MRI images. Considering that the total active cortical area of the alphon extends for a distance of 2 – 3 cm, this implies that *both* hemispheres may contribute to the activity of a single alphon. Thus it is not necessary to view the alphon as a single continuous area of activity. Indeed, the rhythmic coherent activity of an alphon must depend in such a case on long-range communication, perhaps via thalamic connections instead of only cortico-cortical projections.

The findings here are likely to be applicable for interpreting the origin of spontaneous activity in other sensory areas of cortex. Moreover, by carrying out a similar analysis when alpha is suppressed during cognitive studies, it may be possible to identify the precise areas where suppression is strongest by the absence of alphas where formerly they were observed.

This study provides evidence that human alpha rhythm consists of alphas with well-defined locations clustered about the longitudinal fissure and the parieto-occipital sulcus.

# **Part V**

## **Summary**





# Chapter 10

## New algorithms developed

- The SpiderWeb Isocontour and Isosurface algorithms.
- The robot Spider In Web for probing SpiderWeb surfaces.
- The Pointers Best Line and Point Algorithm.
- Three dimensional Delaunay triangularization of arbitrary magnetic field measurements.





# Chapter 11

## Software written

- The CoordAligner software package.
- Forward solution package.
- Inverse solution package.
- PPN Transform Package.
- New Panel Library Graphical User Interface.

### 11.1 Experiments conducted

- Pointers and Spider In Web registration fidelity confirmed by experiment.
- Alphon localization study.
- Inverse solution test.



## Chapter 12

# Concluding Discussion

In this dissertation, I have solved a broad class of problems in basic computational geometry and functional anatomy. I have gone from constructing valid surfaces in volume data to registering those surfaces with magnetic sources. I have built a software system that enables future research into the anatomy and meaning of these sources. The next generation of medical imaging and modeling for functional, and dysfunctional anatomy is now possible.





# **Appendix A**

## **Appendix**

## A.1 PPN Headframe C Source Code

This section give the source code of subroutines that calculate the PPN transform matrix. The subroutine takes a list of the three cardinal points, and calculates the transformation matrix that will take all points in the original data set (containing the cardinal points), and rotate and translate the data set into the PPN Headframe.

```
/******  
void Align(void)  
{  
  int i,j;  
  float tmp[3],Origin[3];  
  Matrix Axes,tmp,PPN_XForm;  
  Mark *M;  
  Stack *hs;  
  
  hs=ArenaDirectory→HotCompositePointer→HotStackPointer;  
  if(!hs)  
  {  
    printf("Align(\"_FILE_\"):no hot stack\n");  
    return;  
  }  
  
  M=hs→CardinalMark;  
  
  if(!(M[0].DrawMe && M[1].DrawMe && M[2].DrawMe ))  
  {  
    printf("Align(\"_FILE_\"): "  
      "Min 3 points required, Align not ready\n");  
    return;  
  }  
  
  IdentMat(Axes);  
  MID_POINT(Origin, M[1].v, M[2].v);  
  MAKE_NORMAL_3VECTOR(Axes[0], M[0].v, Origin);  
  MAKE_NORMAL_3VECTOR(tmp, M[1].v, M[2].v);  
  IdentMat(PPN_XForm);  
  Cross3f(Axes[2],Axes[0],tmp);  
  Cross3f(Axes[1],Axes[2],Axes[0]);  
  Invert4dOpt(Axes, Axes);  
  MatMult(PPN_XForm,Axes,PPN_XForm);  
  MatMult(hs→StackRegistrationMatrix,  
    PPN_XForm, hs→StackRegistrationMatrix);  
  
  for(i=0;i<ArenaDirectory→CardinalMarks;i++)  
  {  
    printf("%s moved from %f,%f,%f ",
```



```

        M[i].MarkName,
        M[i].v[0],M[i].v[1],M[i].v[2]);
    DotF(M[i].v,PPN_XForm,M[i].v);
    printf("to %f,%f,%f\n",
        M[i].v[0],M[i].v[1],M[i].v[2]);
    M[i].IsChanged=1;
    M[i].DrawMe=1;
}

WriteOutStackLandMarks(hs);
PropagateStackRegistration(hs);
AlignHotPoints(hs);
printf("Real PPN Alignment completed\n");
}
/*****/

```

## A.2 Best Point and Line C Source Code.

```
/*
This code and the underlying algorithm
were written and developed by Dan
Karron (karron@nyu.edu). You are free
to use it, but please acknowledge me if
you do anything worth while with it. If
it dosent work for you, let me know and
I will try to help. It works quite well
for me and my colleagues here. July
1991.
*/

#include "stdio.h"
#include "stdlib.h"
#include "signal.h"
#include "gl.h"
#include "strings.h"
#include "math.h"
#include "sigfpe.h"
#include "nr.h"
/* requires the Numerical Recipies library in C */
#include "nrutil.h"
#include "BestLinePublic.h"
#include "PrintMatrix.h"
#define SVD_EDIT
/*****
static double RunCalculation(float **a,int m,float *x)
{
/*
Calculate a best solution, then return
with a figure of merit for how good the
solution really is by using the
solution and taking the difference
between the solution and the known
arbitrary line point normal constant
1.0 .
*/
register int i,j;
static int n=2; /* solve in 2d subspace */
double wmax,wmin,condition;
float **u,*w,**v,*b,*c;

u=matrix(1,m,1,n);
v=matrix(1,m,1,n);
w=vector(1,m);
b=vector(1,m);
```

```

c=vector(1,m);

for(i=1;i≤m;i++)
{
    b[i]=1.0; /* point normal plane constant */
    for(j=1;j≤2;j++)
    {
        u[i][j]=a[i][j]; /* don't touch matrix a */
    }
}

#ifdef DEBUG
/* Lets see the input */
Print_Matrix("input  ",1,a,1,m,1,n);
Print_Vector("constants",1,b,1,m);
#endif DEBUG

svdcmp(u,m,n,w,v); /* decompose u into u,v, and w */

wmax=0.0; /* edit diagonal which is vector v */
for(i=1;i≤m;i++)
{
    if(w[i] > wmax)
    {
        wmax=w[i];
    }
}

#ifdef DEBUG
printf("wmax=%f\n",wmax);
#endif DEBUG

wmin=wmax*(1.0e-6);
for(i=1;i≤m;i++)
{
    if(w[i] < wmin)
    {
        w[i]=0.0;
    }
}

#ifdef DEBUG
printf("wmin=%f\n",wmin);
#endif DEBUG

svbksb(u,w,v,m,n,b,x);
/* svd back substitute to solve for best x */

```



```

#ifdef DEBUG
Print_Matrix("lhs",1,u,1,m,1,n);
Print_Vector("diagonal",1,w,1,m);
Print_Matrix("rhs",1,v,1,m,1,n);
Print_Vector("solutions",1,x,1,n);
#endif DEBUG

nrdot(a,1,m,1,n,x,c);

#ifdef DEBUG
Print_Vector("test solution(c==b?)",1,c,1,m);
#endif DEBUG

condition=0.0;
/* this is not really condition, just an figure of merit
 * for the absolute error, as absolute difference from 1.0
 */

for(i=1;i≤m;i++)
{
    if((c[i]-1.0)<0.0)
        condition -= (c[i]-1.0);
    else
        condition += (c[i]-1.0);
}

#ifdef DEBUG
printf("condition=%f\n",condition);
#endif DEBUG

free_matrix(u,1,m,1,n);
free_matrix(v,1,m,1,n);
free_vector(b,1,m);
free_vector(c,1,m);
free_vector(w,1,n);

return condition;
}
/*****
void NueNogginKnocker(int m,float **plist,float lcoeff[2][3])
{
/*
 * Given a list of m point triples (x,y,z) (which actually should be declared
 * (*p)[3] but I had trouble making that work) return the point normal line
 * coefficients (where the point constant is taken as 1.0, always!)
 * in lcoeff for two point normal planes that describe the line that
 * best fits the list of points in three dimensions.
 */

```

```

register int i;
float **a,*x[4];
double c[4];
int cmaxi;
double cmaxf;
static int n=2;

#ifdef DEBUG
Debug=2;
#endif DEBUG

#ifdef DEBUG
for(i=0;i<m;i++)
/* Recite the incoming point list just to be certain. */
{
    printf("NueNogginKnocker Point [%d] :%f,%f,%f\n",i,
        *(plist[i]), *(plist[i]+1), *(plist[i]+2));
}
#endif DEBUG

if( m < n )
{
    fprintf(stderr,"too few points %d<%d to do anything M<N\n",m,n);
    return;
}

a=matrix(1,m,1,n);
x[1]=vector(1,n); /* hang three solution vectors off x */
x[2]=vector(1,n);
x[3]=vector(1,n);

/* first try */
for(i=0;i<m;i++)
{
    a[i+1][1]= *(plist[i]); /* x */
    a[i+1][2]= *(plist[i]+1);/* y */
}

#ifdef DEBUG
printf("first try\n");
#endif DEBUG

c[1]=RunCalculation(a,m,x[1]);

/* second try */

for(i=0;i<m;i++)

```

```

    {
        a[i+1][1]= *(plist[i]); /* x */
        a[i+1][2]= *(plist[i]+2);/* z */
    }
#ifdef DEBUG
printf("second try\n");
#endif DEBUG

c[2]=RunCalculation(a,m,x[2]);

/* third try */

for(i=0;i<m;i++)
{
    a[i+1][1]= *(plist[i]+1); /* y */
    a[i+1][2]= *(plist[i]+2); /* z */
}

#ifdef DEBUG
printf("third try\n");
#endif DEBUG

c[3]=RunCalculation(a,m,x[3]);

cmaxf=0.0;
/* which is the worst of the three ?,
 * keep the two best ! */

for(i=1;i<=3;i++)
{
    if(cmaxf<c[i])
    {
        cmaxf=c[i];
        cmaxi=i;
    }
}

#ifdef DEBUG
printf("cmax=%f , cmaxi=%d\n",cmaxf,cmaxi);
#endif DEBUG

switch(cmaxi)
/* wire the best two solutions
 * back into the solution for the line
 */
{
    case 3:
        lcoeff[0][0]=x[1][1];

```



```

    lcoeff[0][1]=x[1][2];
    lcoeff[0][2]=0.0;

    lcoeff[1][0]=x[2][1];
    lcoeff[1][1]=0.0;
    lcoeff[1][2]=x[2][2];
    break;

case 2:

    lcoeff[0][0]=x[1][1];
    lcoeff[0][1]=x[1][2];
    lcoeff[0][2]=0.0;

    lcoeff[1][0]=0.0;
    lcoeff[1][1]=x[3][1];
    lcoeff[1][2]=x[3][2];
    break;

case 1:
    lcoeff[0][0]=x[2][1];
    lcoeff[0][1]=0.0;
    lcoeff[0][2]=x[2][2];

    lcoeff[1][0]=0.0;
    lcoeff[1][1]=x[3][1];
    lcoeff[1][2]=x[3][2];
    break;
}

#ifdef DEBUG
printf("lcoeff [0] = %f , %f , %f\n",
    lcoeff[0][0],lcoeff[0][1],lcoeff[0][2]);
printf("lcoeff [1] = %f , %f , %f\n",
    lcoeff[1][0],lcoeff[1][1],lcoeff[1][2]);
#endif

free_matrix(a,1,m,1,n);
free_vector(x[1],1,n);
free_vector(x[2],1,n);
free_vector(x[3],1,n);
}
/*****/
void NitherNoid(int mm,float **llist,float point[3],
    int debug_me)
{
/*
Find the best point to solve a mm long

```

```

list of lines attached to llist.
Actually, the solution is defined the
same as the best point to solve for a
list of lines consisting of pairs of
planes.
*/

register int i,j,k;
float **u,**w,**v,**a,*b,*c,*x,**cvm;
float wmax,wmin;
int n=3; /* solution is in 3 dimensions */
int m=mm*2; /* each line consists of two planes */

#ifdef DEBUG
Debug=1;
#endif

#ifdef DEBUG
if(debug_me)
for(j=0;j<mm;j++)
{
printf("NitherNoid:Line[%d]:%+e,%+e,%+e\n\t%+e,%+e,%+e\n",j,
*(llist[j]+0),*(llist[j]+1),*(llist[j]+2),
*(llist[j]+3),*(llist[j]+4),*(llist[j]+5));
}
#endif

if(m < n )
{
fprintf(stderr,"Too few plane pairs to solve for M<N\n");
return;
}

a=matrix(1,m,1,n);
b=vector(1,m);
c=vector(1,m);
u=matrix(1,m,1,n);
v=matrix(1,m,1,n);
cvm=matrix(1,m,1,n);
w=vector(1,m);
x=vector(1,n);

for(i=0;i<mm;i++)
{
b[i*2+1]=b[i*2+2]=1.0;
/* point normal plane constant */

u[i*2+1][1]=a[i*2+1][1]=*(llist[i]+0);

```

```

    u[i*2+1][2]=a[i*2+1][2]= *(llist[i]+1);
    u[i*2+1][3]=a[i*2+1][3]= *(llist[i]+2);

    u[i*2+2][1]=a[i*2+2][1]= *(llist[i]+3);
    u[i*2+2][2]=a[i*2+2][2]= *(llist[i]+4);
    u[i*2+2][3]=a[i*2+2][3]= *(llist[i]+5);
}

#ifdef DEBUG
Print_Matrix("NitherNoid:input",1,a,1,m,1,n);
Print_Vector("NitherNoid:constants",1,b,1,m);
#endif DEBUG

svdcmp(u,m,n,w,v);

#ifdef SVD_EDIT
wmax=0.0;
for(i=1;i≤m;i++)
{
    if(w[i] > wmax)
    {
        wmax=w[i];
    }
}
wmin=wmax*(1.0e-6);
for(i=1;i≤m;i++)
{
    if(w[i] < wmin)
    {
        w[i]=0.0;
    }
}
#endif SVD_EDIT

svbksb(u,w,v,m,n,b,x);

#ifdef DEBUG
Print_Matrix("lhs",1,u,1,m,1,n);
Print_Vector("diagonal",1,w,1,m);
Print_Matrix("rhs",1,v,1,m,1,n);
Print_Vector("NitherNoid solution",1,x,1,n);
nrldot(a,1,m,1,n,x,c);
Print_Vector("NitherNoid test solution(c==b?)",1,c,1,m);
#endif DEBUG

point[0]=x[1];
point[1]=x[2];
point[2]=x[3];

```



```
free_matrix(a,1,m,1,n);
free_vector(b,1,m);
free_vector(c,1,m);
free_matrix(u,1,m,1,n);
free_matrix(v,1,m,1,n);
free_vector(w,1,n);
free_vector(x,1,n);
free_matrix(cvm,1,m,1,n);
}
/*****/
```

## A.3 Best Sphere C Source Code

```
#include <gl.h>
#include <math.h>
#include <stdlib.h>

#include "nrutil.h"
#include "nr.h"

#include "Arena.h"
#include "MathPublic.h"
#include "SharedProcsPublic.h"
#include "SphereOpsPublic.h"

#include "PrintMatrix.h"
/* A collection of macros that will print
 * formatted matrices and vectors if DEBUG is defined */

static call=0;

/*****
static float SumSquaredDist(float *x)
{
/* This subroutine calculates the distance between */
/* each trial point and the estimated sphere origin */
/* and radius. */
/* */
/* Is it used by the amoeba to slide to a better and */
/* better answer */

register i;
float dt;
float d=0.0;

for(i=0;i<n;i++)
{
dt=DistSqf(fpl[i][0],fpl[i][1],fpl[i][2],x[1],x[2],x[3]);
#ifdef DEBUG
if(getenv("DEBUG"))
printf("%d: (%f-%f)^2+ (%f-%f)^2+ (%f-%f)^2=%f ,R^2=%f ,r^2-R^2=%f\n",i,
fpl[i][0],x[1],fpl[i][1],x[2],fpl[i][2],x[3],
dt,x[4]*x[4],dt-x[4]*x[4]);
#endif
dt -= x[4]*x[4];
if(dt<0)
dt= -dt;
d += dt;
}
}
```

```

#ifdef DEBUG
if(getenv("DEBUG"))
    printf("call %d with(%f,%f,%f,%f) has d=%f\n",call,x[1],x[2],x[3],x[4],d);
call++;
#endif

return d;
}
/*****/
void NewFindSphere(int npoints,float pl[][3],float param[4])
{
    /* Given a list of points, find the best fitting sphere. */
    /* npoints is the number of points passed in the */
    /* list of pointers to float triples of points to fit */
    /* param is the origin and radius of the sphere that */
    /* best solves the list of points. */
    register i,j;
    float *y;
    float **p;
    int nfunk=0;
    static float starter_guess[5][4]={
        {-2.01, 0.0, 4.02, 9.0 },
        { 0.01, 0.02, 4.03, 10.1 },
        { 0.008,0.05, 4.090, 10.59 },
        { 0.01, 0.044, 4.03, 9.0 },
        { 0.023,0.046, 3.740, 8.8345 }};
    Debug=0;
    call=0;
    n=npoints;
    new_fpl=pl;

    p=matrix(1,5,1,4);
    y=vector(1,5);

    for(i=0;i<5;i++)
        for(j=0;j<4;j++)
            p[i+1][j+1]=starter_guess[i][j];

    Print_Matrix("Fit Point List",1,new_fpl,0,n-1,0,2);

    Print_Matrix("Simplex Starter",1,p,1,5,1,4);

    y[1]=NewSumSquaredDist(p[1]);
    y[2]=NewSumSquaredDist(p[2]);
    y[3]=NewSumSquaredDist(p[3]);
    y[4]=NewSumSquaredDist(p[4]);
    y[5]=NewSumSquaredDist(p[5]);

```



```

Print_Vector("Starter Solution",1,y,1,5);

amoeba(p,y,4,NewSumSquaredDist,&nfunk);

for(i=0;i<4;i++)
    param[i]=p[1][i+1];

if(Debug)
{
    printf("%d iterations\n",nfunk);
    Print_Vector("Simplex func solutions",1,y,1,5);
    Print_Matrix("Simplex Values",1,p,1,5,1,4);
    printf("R=%f\n",param[3]*param[3]);
    for(i=0;i<n;i++)
    {
        printf("solution for point %d:%f\n",i,
            (new_fpl[i][0]-param[0])*(new_fpl[i][0]-param[0])+
            (new_fpl[i][1]-param[1])*(new_fpl[i][1]-param[1])+
            (new_fpl[i][2]-param[2])*(new_fpl[i][2]-param[2]));
    }
}
free_matrix(p,1,5,1,4);
free_vector(y,1,4);
}
/*****
void CalculateBestSphere(void)
{
    /* Assemble a list of point from each slice in a stack */
    /* to calculate the best sphere origin and radius. */
    register int i,j;
    int point_count=0;
    float **point_list=NULL;
    Stack *hsp=ArenaDirectory→HotCompositePointer→HotStackPointer;

    if(!hsp)
        return;

    for(i=0;i<SPHERE_POINTS;i++) /* go through the entire composite, not just hot stack */
    {
        for(j=0;j<hsp→Slices;j++)
        {
            if(hsp→Slice[j].SpherePixels[i][0] ≥ 0 )
            {
                point_count++;
                if(point_list)
                    point_list=realloc(point_list,point_count*sizeof(float *));
                else

```

```

        point_list=calloc(point_count,sizeof(float *));
        point_list[point_count-1]=
        &hsp→Slice[j].SpherePoints[i][0];
    }
}

for(i=0;i<point_count;i++)
    printf("Calculate Sphere With:=%d:%f,%f,%f\n",i,
    point_list[i][0],
    point_list[i][1],
    point_list[i][2]);

if(point_count>ArenaDirectory→BSphere→CalculateSphereThreshold)
{
    NewFindSphere(point_count,point_list,ArenaDirectory→BSphere→Sphere.param);
    printf("Sphere: (x - %5.2f)^2 + (y - %5.2f)^2 + (z - %5.2f)^2 = %5.2f^2 \n",
    ArenaDirectory→BSphere→Sphere.param[0],
    ArenaDirectory→BSphere→Sphere.param[1],
    ArenaDirectory→BSphere→Sphere.param[2],
    ArenaDirectory→BSphere→Sphere.param[3]);
    ArenaDirectory→BSphere→Sphere.DrawMe=1;
}
if(point_list)
    free(point_list);
}
/*****/

```

## A.4 Sample Pointer Error Report For Analyzing Pointer Images

The next section shows an example of an error report generated automatically in  $\text{\LaTeX}$  by the CoordAligner program that details all of the errors in constructing each line from a list of points, and the skew error of the list of converging lines.

Each line is constructed from a list of points that do not lie precisely in a straight line. The Pointers algorithm calculates the best possible line through the points. This report gives the distance from each constituent point to this best line. The lines should converge to a point located at the apex of the Pointers devices, and directly over a cardinal landmark. However, these lines do not converge to a point, and we need to find the point that is located such that its distance is minimized to each line. This distance, from this point to each line, is the skew error.

These error distances shown in the report for each constructed cardinal landmark point, and for each constructed line.



## Pointer Algorithm Line and Point Error Report.

### POINTER Detailed Stack 1001(0) Pointer Analysis

#### POINTER Lines Skewering Cardinal Mark “Nmri0.1001”

| Index | Line Name<br>N Points    | A<br>D         | B<br>E          | C<br>F          | Skew Distance<br>(cm) |
|-------|--------------------------|----------------|-----------------|-----------------|-----------------------|
| 0     | “Nmri0.1001.LINE.0”<br>5 | 0.098<br>0.156 | -0.059<br>0.000 | 0.000<br>-0.096 | 0.085                 |
| 1     | “Nmri0.1001.LINE.1”<br>5 | 0.102<br>0.032 | -0.099<br>0.000 | 0.000<br>0.092  | 0.099                 |
| 2     | “Nmri0.1001.LINE.2”<br>3 | 0.078<br>0.205 | 0.119<br>0.000  | 0.000<br>-0.171 | 0.095                 |
| 3     | “Nmri0.1001.LINE.3”<br>3 | 0.077<br>0.010 | 0.111<br>0.000  | 0.000<br>0.129  | 0.251                 |

#### Point Construction of Line “Nmri0.1001.LINE.0”

Line is constructed from 5 points, and passes 0.085 cm from landmark “Nmri0.1001”.

| Index | Point Name                  | X      | Y     | Z     | Err   |
|-------|-----------------------------|--------|-------|-------|-------|
| 0     | “Nmri0.1001.LINE.0.POINT.0” | 11.716 | 2.600 | 8.519 | 0.017 |
| 1     | “Nmri0.1001.LINE.0.POINT.1” | 11.567 | 2.300 | 8.328 | 0.056 |
| 2     | “Nmri0.1001.LINE.0.POINT.2” | 11.358 | 2.000 | 7.947 | 0.010 |
| 3     | “Nmri0.1001.LINE.0.POINT.3” | 11.099 | 1.700 | 7.510 | 0.119 |
| 4     | “Nmri0.1001.LINE.0.POINT.4” | 11.049 | 1.400 | 7.499 | 0.086 |

The total of the error distances for this list of points with respect to the line is 0.288, and the mean error distance is 0.058

#### Point Construction of Line “Nmri0.1001.LINE.1”

Line is constructed from 5 points, and passes 0.099 cm from landmark “Nmri0.1001”.

| Index | Point Name                  | X      | Y     | Z     | Err   |
|-------|-----------------------------|--------|-------|-------|-------|
| 0     | "Nmri0.1001.LINE.1.POINT.0" | 12.314 | 2.600 | 6.524 | 0.032 |
| 1     | "Nmri0.1001.LINE.1.POINT.1" | 11.996 | 2.300 | 6.703 | 0.043 |
| 2     | "Nmri0.1001.LINE.1.POINT.2" | 11.716 | 2.000 | 6.815 | 0.055 |
| 3     | "Nmri0.1001.LINE.1.POINT.3" | 11.368 | 1.700 | 6.782 | 0.096 |
| 4     | "Nmri0.1001.LINE.1.POINT.4" | 11.178 | 1.400 | 6.972 | 0.035 |

The total of the error distances for this list of points with respect to the line is 0.260, and the mean error distance is 0.052

### Point Construction of Line "Nmri0.1001.LINE.2"

Line is constructed from 3 points, and passes 0.095 cm from landmark "Nmri0.1001".

| Index | Point Name                  | X      | Y     | Z     | Err   |
|-------|-----------------------------|--------|-------|-------|-------|
| 0     | "Nmri0.1001.LINE.2.POINT.0" | 11.188 | 1.100 | 7.566 | 0.013 |
| 1     | "Nmri0.1001.LINE.2.POINT.1" | 11.587 | 0.800 | 8.037 | 0.026 |
| 2     | "Nmri0.1001.LINE.2.POINT.2" | 12.105 | 0.500 | 8.665 | 0.012 |

The total of the error distances for this list of points with respect to the line is 0.051, and the mean error distance is 0.017

### Point Construction of Line "Nmri0.1001.LINE.3"

Line is constructed from 3 points, and passes 0.251 cm from landmark "Nmri0.1001".

| Index | Point Name                  | X      | Y     | Z     | Err   |
|-------|-----------------------------|--------|-------|-------|-------|
| 0     | "Nmri0.1001.LINE.3.POINT.0" | 11.358 | 1.100 | 6.916 | 0.024 |
| 1     | "Nmri0.1001.LINE.3.POINT.1" | 11.906 | 0.800 | 6.827 | 0.060 |
| 2     | "Nmri0.1001.LINE.3.POINT.2" | 12.225 | 0.500 | 6.871 | 0.037 |

The total of the error distances for this list of points with respect to the line is 0.121, and the mean error distance is 0.040

## Lines Skewering Cardinal Mark “LPAmri1.1001”

| Index | Line Name<br>N Points      | A<br>D          | B<br>E         | C<br>F          | Skew Distance<br>(cm) |
|-------|----------------------------|-----------------|----------------|-----------------|-----------------------|
| 0     | “LPAmri1.1001.LINE.0”<br>4 | -1.458<br>0.000 | 0.128<br>0.178 | 0.000<br>-0.596 | 0.096                 |
| 1     | “LPAmri1.1001.LINE.1”<br>4 | 0.195<br>0.086  | 0.132<br>0.000 | 0.000<br>2.199  | 0.041                 |
| 2     | “LPAmri1.1001.LINE.2”<br>5 | 0.409<br>0.000  | 0.131<br>0.123 | 0.000<br>0.143  | 0.066                 |
| 3     | “LPAmri1.1001.LINE.3”<br>4 | -0.289<br>0.000 | 0.132<br>0.111 | 0.000<br>0.309  | 0.058                 |

### Point Construction of Line “LPAmri1.1001.LINE.0”

Line is constructed from 4 points, and passes 0.096 cm from landmark “LPAmri1.1001”.

| Index | Point Name                    | X     | Y     | Z     | Err   |
|-------|-------------------------------|-------|-------|-------|-------|
| 0     | “LPAmri1.1001.LINE.0.POINT.0” | 0.140 | 9.500 | 1.144 | 0.010 |
| 1     | “LPAmri1.1001.LINE.0.POINT.1” | 0.100 | 9.200 | 1.077 | 0.025 |
| 2     | “LPAmri1.1001.LINE.0.POINT.2” | 0.090 | 8.900 | 0.953 | 0.019 |
| 3     | “LPAmri1.1001.LINE.0.POINT.3” | 0.100 | 8.600 | 0.897 | 0.035 |

The total of the error distances for this list of points with respect to the line is 0.088, and the mean error distance is 0.022

### Point Construction of Line “LPAmri1.1001.LINE.1”

Line is constructed from 4 points, and passes 0.041 cm from landmark “LPAmri1.1001”.

| Index | Point Name                    | X      | Y     | Z     | Err   |
|-------|-------------------------------|--------|-------|-------|-------|
| 0     | “LPAmri1.1001.LINE.1.POINT.0” | -1.315 | 9.500 | 0.516 | 0.011 |
| 1     | “LPAmri1.1001.LINE.1.POINT.1” | -1.105 | 9.200 | 0.494 | 0.005 |
| 2     | “LPAmri1.1001.LINE.1.POINT.2” | -0.896 | 8.900 | 0.471 | 0.020 |
| 3     | “LPAmri1.1001.LINE.1.POINT.3” | -0.707 | 8.600 | 0.494 | 0.013 |

The total of the error distances for this list of points with respect to the line is 0.048, and the mean error distance is 0.012



### Point Construction of Line “LPAmri1.1001.LINE.2”

Line is constructed from 5 points, and passes 0.066 cm from landmark “LPAmri1.1001”.

| Index | Point Name                    | X      | Y     | Z      | Err   |
|-------|-------------------------------|--------|-------|--------|-------|
| 0     | “LPAmri1.1001.LINE.2.POINT.0” | -0.587 | 9.500 | -1.210 | 0.034 |
| 1     | “LPAmri1.1001.LINE.2.POINT.1” | -0.518 | 9.200 | -0.885 | 0.028 |
| 2     | “LPAmri1.1001.LINE.2.POINT.2” | -0.408 | 8.900 | -0.616 | 0.030 |
| 3     | “LPAmri1.1001.LINE.2.POINT.3” | -0.278 | 8.600 | -0.336 | 0.050 |
| 4     | “LPAmri1.1001.LINE.2.POINT.4” | -0.238 | 8.300 | -0.224 | 0.067 |

The total of the error distances for this list of points with respect to the line is 0.209, and the mean error distance is 0.042

### Point Construction of Line “LPAmri1.1001.LINE.3”

Line is constructed from 4 points, and passes 0.058 cm from landmark “LPAmri1.1001”.

| Index | Point Name                    | X     | Y     | Z      | Err   |
|-------|-------------------------------|-------|-------|--------|-------|
| 0     | “LPAmri1.1001.LINE.3.POINT.0” | 0.878 | 9.500 | -0.179 | 0.008 |
| 1     | “LPAmri1.1001.LINE.3.POINT.1” | 0.728 | 9.200 | -0.078 | 0.006 |
| 2     | “LPAmri1.1001.LINE.3.POINT.2” | 0.589 | 8.900 | 0.023  | 0.013 |
| 3     | “LPAmri1.1001.LINE.3.POINT.3” | 0.469 | 8.600 | 0.146  | 0.011 |

The total of the error distances for this list of points with respect to the line is 0.037, and the mean error distance is 0.009

## Lines Skewering Cardinal Mark “RPAmri2.1001”

| Index | Line Name<br>N Points      | A<br>D          | B<br>E           | C<br>F          | Skew Distance<br>(cm) |
|-------|----------------------------|-----------------|------------------|-----------------|-----------------------|
| 0     | “RPAmri2.1001.LINE.0”<br>5 | -0.568<br>0.000 | -0.238<br>-0.168 | 0.000<br>-0.396 | 0.049                 |
| 1     | “RPAmri2.1001.LINE.1”<br>4 | 0.313<br>0.281  | -0.083<br>0.000  | 0.000<br>1.379  | 0.039                 |
| 2     | “RPAmri2.1001.LINE.2”<br>5 | 0.760<br>0.000  | 0.000<br>-0.125  | 0.100<br>0.164  | 0.120                 |
| 3     | “RPAmri2.1001.LINE.3”<br>6 | -0.183<br>0.000 | -0.171<br>-0.114 | 0.000<br>0.448  | 0.057                 |

### Point Construction of Line “RPAmri2.1001.LINE.0”

Line is constructed from 5 points, and passes 0.049 cm from landmark “RPAmri2.1001”.

| Index | Point Name                    | X     | Y      | Z     | Err   |
|-------|-------------------------------|-------|--------|-------|-------|
| 0     | “RPAmri2.1001.LINE.0.POINT.0” | 1.678 | -8.200 | 0.959 | 0.009 |
| 1     | “RPAmri2.1001.LINE.0.POINT.1” | 1.791 | -8.500 | 1.079 | 0.004 |
| 2     | “RPAmri2.1001.LINE.0.POINT.2” | 1.920 | -8.800 | 1.225 | 0.015 |
| 3     | “RPAmri2.1001.LINE.0.POINT.3” | 2.030 | -9.100 | 1.303 | 0.031 |
| 4     | “RPAmri2.1001.LINE.0.POINT.4” | 2.181 | -9.400 | 1.475 | 0.014 |

The total of the error distances for this list of points with respect to the line is 0.074, and the mean error distance is 0.015

### Point Construction of Line “RPAmri2.1001.LINE.1”

Line is constructed from 4 points, and passes 0.039 cm from landmark “RPAmri2.1001”.

| Index | Point Name                    | X     | Y      | Z     | Err   |
|-------|-------------------------------|-------|--------|-------|-------|
| 0     | “RPAmri2.1001.LINE.1.POINT.0” | 0.910 | -8.500 | 0.545 | 0.033 |
| 1     | “RPAmri2.1001.LINE.1.POINT.1” | 0.894 | -8.800 | 0.552 | 0.030 |
| 2     | “RPAmri2.1001.LINE.1.POINT.2” | 0.794 | -9.100 | 0.541 | 0.027 |
| 3     | “RPAmri2.1001.LINE.1.POINT.3” | 0.692 | -9.400 | 0.592 | 0.014 |

The total of the error distances for this list of points with respect to the line is 0.104, and the mean error distance is 0.026

### Point Construction of Line “RPAmri2.1001.LINE.2”

Line is constructed from 5 points, and passes 0.120 cm from landmark “RPAmri2.1001”.

| Index | Point Name                    | X     | Y      | Z      | Err   |
|-------|-------------------------------|-------|--------|--------|-------|
| 0     | “RPAmri2.1001.LINE.2.POINT.0” | 1.329 | -8.200 | -0.206 | 0.034 |
| 1     | “RPAmri2.1001.LINE.2.POINT.1” | 1.359 | -8.500 | -0.352 | 0.035 |
| 2     | “RPAmri2.1001.LINE.2.POINT.2” | 1.432 | -8.800 | -0.602 | 0.038 |
| 3     | “RPAmri2.1001.LINE.2.POINT.3” | 1.422 | -9.100 | -0.882 | 0.026 |
| 4     | “RPAmri2.1001.LINE.2.POINT.4” | 1.452 | -9.400 | -1.073 | 0.009 |

The total of the error distances for this list of points with respect to the line is 0.142, and the mean error distance is 0.028

### Point Construction of Line “RPAmri2.1001.LINE.3”

Line is constructed from 6 points, and passes 0.057 cm from landmark “RPAmri2.1001”.

| Index | Point Name                    | X     | Y      | Z      | Err   |
|-------|-------------------------------|-------|--------|--------|-------|
| 0     | “RPAmri2.1001.LINE.3.POINT.0” | 2.026 | -7.900 | 0.186  | 0.075 |
| 1     | “RPAmri2.1001.LINE.3.POINT.1” | 2.126 | -8.200 | 0.175  | 0.064 |
| 2     | “RPAmri2.1001.LINE.3.POINT.2” | 2.465 | -8.500 | 0.074  | 0.020 |
| 3     | “RPAmri2.1001.LINE.3.POINT.3” | 2.787 | -8.800 | 0.026  | 0.032 |
| 4     | “RPAmri2.1001.LINE.3.POINT.4” | 3.066 | -9.100 | -0.087 | 0.011 |
| 5     | “RPAmri2.1001.LINE.3.POINT.5” | 3.325 | -9.400 | -0.165 | 0.012 |

The total of the error distances for this list of points with respect to the line is 0.213, and the mean error distance is 0.036

END OF  
Pointer Algorithm Line and Point Error Report.





# Bibliography

- [1] O. C. Zienkiewicz, *The Finite Element Method*. London: McGraw-Hill Book Company (UK) Limited, third ed., 1977.
- [2] T. D. Bui and V. T. N. Hanh, "Mesh Generation in 3-D for Finite Element Analysis," in *Progress In Computer Graphics* (G. W. Zobrist and C. Sabharwal, eds.), no. 1 in Progress in Computer Graphics, ch. 7, pp. 254–281, 355 Chestnut Street, Norwood, New Jersey 07648: Ablex Publishing Corporation, 1992.
- [3] H. E. Cline, W. E. Lorensen, S. Ludke, and B. C. T. C. R. Crawford, "Two Algorithms for the Reconstruction of Surfaces from Tomograms," *Medical Physics*, vol. 15, pp. 320–327, May 1988.
- [4] A. Kriete, S. Rohrbach, T. Schwebel, H.-J. Wagner, and U. Behrens, "Data representation and visualization in 4-D microscopy," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, (Chapel Hill, North Carolina), pp. 396–409, The Society of Photo-Optical Instrumentation Engineers, SPIE, Oct. 1992.
- [5] D. Meyers, S. Skinner, and K. Sloan, "Surfaces from Contours," *ACM Transactions on Graphics*, vol. 11, pp. 228–258, July 1992.
- [6] D. Karron, "How to Lie with Isosurfaces." Proceedings of the Visualization 92 Conference, How to Lie in Scientific Visualization, Oct. 1992.
- [7] Xante Corporation, "Xante Accel-A-Writer: Volume II for HP LaserJet Printers," technical users manual, Xante Corporation, 1992.
- [8] Polhemus Navigation, Inc., "3Space<sup>TM</sup> User's Manual," Technical Users Manual 4A, Polhemus Navigation Inc., P.O. Box 560, Hercules Drive, Colchester, Vermont, 05446, 1986.
- [9] M. R. Stytz, G. Frieder, and O. Frieder, "Three-Dimensional Medical Imaging: Algorithms and Computer Systems," *ACM Computing Surveys*, vol. 23, pp. 421–499, Dec. 1991.

- [10] E. Keppel, "Approximating Complex Surfaces by Triangularization of Contour Lines," *IBM Journal of Research and Development*, vol. 19, pp. 2–11, Jan. 1975.
- [11] W. M. Snyder, "Contour Plotting," *ACM Transactions on Mathematical Software*, vol. 4, no. 3, pp. 290–294, 1978.
- [12] H. Fuchs, Z. M. Kedem, and S. P. Uelson, "Optimal Surface Reconstruction from Planar Contours," *Communications of the ACM*, vol. 20, pp. 693–702, Oct. 1977.
- [13] Brigham Young University Computer Science Department, "Movie BYU<sup>TM</sup> users manual." Computer Software Users Manual with Software, 1986.
- [14] H. N. Christiansen and T. W. Sederberg, "Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics," *Computer Graphics, Proceedings of SIGGRAPH*, vol. 12, pp. 187–192, Aug. 1978.
- [15] P. Dev, S. Wood, D. N. White, S. W. Young, and J. P. Duncan, "An Interactive Graphics System for Planning Reconstructive Surgery," in *Proceedings of the Fourth NCGA*, pp. 130–135, 1983.
- [16] P. Dev, L. Fellingham, A. Vassiliadis, S. T. Woolson, D. White, and S. L. Young, "3D Graphics for Interactive Surgical Simulation and Implant Design," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, pp. 52–57, 1984.
- [17] D. J. Meagher, "Applying Solids Processing Methods to Medical Planning," in *Proceedings of NCGA 1985*, pp. 101–109, Apr. 1985.
- [18] S. L. Wood, "Visualization and Modeling of 3-D Structures," *IEEE Engineering in Medicine and Biology*, vol. 11, pp. 72–79, June 1992.
- [19] M. J. Zyda, A. R. Jones, and P. G. Hogan, "Surface Construction from Plane Contours," *Computers and Graphics*, vol. 11, pp. 393–408, Apr. 1987.
- [20] M. Hall, "Defining surfaces from contour data," in *Graphics Gems* (A. S. Glassner, ed.), ch. 10, pp. 558–561, Boston: Academic Press, Inc., 1990.
- [21] T. Feldman, "Generating Isovalue Contours from a Pixmap," in *Graphics Gems III* (D. Kirk, ed.), vol. 3, ch. 1, pp. 29–33, Boston: Academic Press, Inc., 1992.
- [22] B. S. Morse, W. A. Barrett, J. K. Udupa, and R. P. Burton, "Trainable Optimal Boundary Finding Using two-dimensional Dynamic Programming," Technical



- Report MIPG180, Medical Image Processing Group, University Of Pennsylvania, Medical Image Processing Group, Department of Radiology, University Of Pennsylvania, Brockley Hall, Fourth Floor, 418 Service Drive, Philadelphia, Pennsylvania, 19104-6021, Mar. 1991.
- [23] J. K. Udupa, S. Samarasekera, and W. A. Barrett, "Boundary detection via dynamic programming," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, pp. 33-39, Chapel Hill, North Carolina: SPIE, Oct. 1992.
  - [24] J. K. Udupa, S. Samarasekera, R. A. Lotufo, G. T. Herman, B. E. Hirsch, W. A. Barrett, and R. J. Gonclaves, "Boundary Detection, Kinematic Analysis and Shape-Based Interpolation," Technical Report MIPG187, Medical Image Processing Group, University Of Pennsylvania, Medical Image Processing Group, Department of Radiology, University Of Pennsylvania, Brockley Hall, Fourth Floor, 418 Service Drive, Philadelphia, Pennsylvania, 19104-6021, Aug. 1992.
  - [25] Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien, "Surface Coding Based on Morse Theory," *IEEE Computer Graphics and Applications*, vol. 11, pp. 66-78, July 1991.
  - [26] Y. Shinagawa and T. L. Kunii, "Constructing a Reeb Graph Automatically from Cross Sections," *IEEE Computer Graphics and Applications*, vol. 11, pp. 44-51, Nov. 1991.
  - [27] M. W. Vannier, J. L. Marsh, and J. O. Warren, "Three Dimensional Computer Graphics for Craniofacial Surgical Planning and Evaluation," *ACM Computer Graphics*, vol. 17, pp. 408-418, July 1983.
  - [28] C. Cutting, B. Grayson, F. Bookstein, L. Fellingham, and J. G. McCarthy, "Computer-Aided Planning and Evaluation of Facial and Orthognathic Surgery," *Clinics in Plastic Surgery*, vol. 13, pp. 449-462, July 1986.
  - [29] C. Cutting, F. Bookstein, B. Grayson, and J. G. McCarthy, "Computer aided planning of orthognathic surgery," *Plastic Surgery Research Council Meetings Abstracts*, Apr. 1984.
  - [30] C. B. Cutting, "Applications of Computer Graphics to the Evaluation and Treatment of Major Craniofacial Malformations," in *3D Imaging in Medicine* (J. K. Udupa and G. T. Herman, eds.), chapter 6, pp. 163-190, Boca Raton, Florida: CRC Press, Inc., 1991.
  - [31] G. T. Herman and J. K. Udupa, "Display of 3-D Digital Images: Computational Foundations and Medical Applications," *IEEE Computer Graphics and Applications*, vol. 3, pp. 39-46, Aug. 1983.

- [32] L. J. Brewster, S. S. Trivedi, J. K. Tuy, and J. K. Udupa, "Interactive Surgical Planning," *IEEE Computer Graphics and Applications*, vol. 4, pp. 391–400, Mar. 1984.
- [33] S. M. Goldwasser, R. A. Reynolds, T. Bapty, D. Baraff, J. Summers, D. A. Talton, and E. Walsh, "Physician's workstation with real-time performance," *IEEE Computer Graphics and Applications*, vol. 5, pp. 44–56, Dec. 1985.
- [34] J. K. Udupa and D. Odhner, "Fast Visualization, Manipulation, and Analysis of Binary Volumetric Objects," *IEEE Computer Graphics and Applications*, vol. 11, pp. 53–62, Nov. 1991.
- [35] R. Gilchrist, "A Multi-Mode 3D Renderer for Clinical and Scientific Visualization," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, (Chapel Hill, North Carolina), pp. 362–373, The Society of Photo-Optical Instrumentation Engineers, SPIE, Oct. 1992.
- [36] P. Heffernan and D. Dekel, "Imaging Applications Platform: Concept to implementation," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, pp. 495–509, Chapel Hill, North Carolina: SPIE, Oct. 1992.
- [37] L. T. S. Leung and W. Synnott, "G2: The design and realization of an accelerator for volume visualization," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, (Chapel Hill, North Carolina), pp. 384–395, The Society of Photo-Optical Instrumentation Engineers, SPIE, Oct. 1992.
- [38] D. Jackèl and W. Strasser, "Reconstructing Solids from Tomographic Scans—The PARCUM II System—," *Advances In Computer Graphics Hardware II*, vol. 2, pp. 101–109, 1988.
- [39] J.-P. Thirion, "Segmentation of Tomographic Data Without Image Reconstruction," *IEEE Transactions on Medical Imaging*, vol. 11, pp. 102–110, Mar. 1992.
- [40] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. New York, New York: IEEE Press, 1988.
- [41] G. T. Herman and H. K. Liu, *Three-Dimensional Display of Human Organs from Computed Tomograms*, pp. 1–21. New York, New York: Academic Press, Inc., 1979.
- [42] H. Gouraud, "Continuous shading of curved surfaces," *IEEE Transactions on Computers*, vol. 20, pp. 623–629, July 1971.



- [43] Lih-Shyang, G. T. Herman, R. A. Reynolds, and J. K. Udupa, "Surface Shading in the Cuberille Environment," *IEEE Computer Graphics and Applications*, vol. 5, pp. 33–43, Dec. 1985.
- [44] E. Artzy, G. Frieder, and G. T. Herman, "The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm", journal = "Computer graphics and Image Processing," vol. 15, pp. 1–24, Jan. 1981.
- [45] D. Gordon and R. A. Reynolds, "Image Space Shading of 3-Dimensional Objects," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 361–376, 1985.
- [46] S. Bright and S. Laffin, "Shading of Solid Voxel Models," *Computer Graphics Forum*, vol. 5, pp. 131–138, June 1986.
- [47] D. Cohen, A. Kaufman, R. Bakalash, and S. Bergman, "Real time discrete shading," *The Visual Computer*, vol. 6, pp. 16–27, Feb. 1990.
- [48] A. Kaufman, "Efficient Algorithms for 3D Scan-Conversion of Parametric Curves, Surfaces, and Volumes," *ACM SIGGRAPH*, vol. 21, pp. 171–179, July 1987.
- [49] A. Kaufman, R. Bakalash, D. Cohen, and R. Yagel, "Introduction to Chapter 6: Architectures for Volume Rendering," in *Volume Visualization* (A. Kaufman, ed.), IEEE Computer Society Press Tutorial, pp. 311–320, Los Alamitor, California: IEEE Computer Society Press, 1991.
- [50] K. H. Höne, M. Bomans, A. Pommert, M. Riemer, C. Schiers, U. Tiede, and G. Wiebecke, "3D Visualization of tomographic volume data using the generalized voxel model," *The Visual Computer*, vol. 6, pp. 28–36, Feb. 1990.
- [51] R. E. Webber, "Ray tracing voxel based data via biquadratic local surface interpolation," *The Visual Computer*, vol. 6, pp. 8–15, Feb. 1990.
- [52] D. C. Hemmy and P. L. Tessier, "CT of Dry Skulls with Craniofacial Deformities: Accuracy of Three-Dimensional Reconstruction," *Radiology*, vol. 157, pp. 113–116, Oct. 1985.
- [53] H. Rusinek, M. E. Noz, G. Q. M. Jr., A. Kalvin, B. Haddad, D. Dean, and C. Cutting, "Quantitative and Qualitative Comparison of Volumetric and Surface Rendering Techniques," *IEEE Transactions on Nuclear Science*, vol. 38, pp. 659–662, Apr. 1991.



- [54] A. D. Kalvin, *Segmentation and Surface-Based Modeling of Objects in Three-Dimensional Biomedical Images*. Ph.D. dissertation, New York University, Computer Science Department, 1991.
- [55] Pixar Inc., "ChapVolume Users Manual," tech. rep., Pixar, Inc., 1985.
- [56] S. W. Zucker and R. A. Hummel, "A Three-Dimensional Edge Operator," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, pp. 244–251, May 1981.
- [57] R. A. Hummel and S. W. Zucker, "On the Foundations of Relaxation Labeling Processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 267–283, 1983.
- [58] D. G. Morgenthaler and A. Rosenfeld, "Surfaces in Three-Dimensional Digital Images," *Information and Control*, vol. 51, pp. 227–247, July 1981.
- [59] G. Wyvill, C. McPheeters, and B. Wyvill, "Data Structure for Soft Objects," *Visual Computer*, vol. 2, no. 4, pp. 227–234, 1986.
- [60] A. V. Gelder and J. Wilhelms, "Topological Ambiguities in Isosurface Generation," tech. rep., University of California at Santa Cruz, 1990.
- [61] H. H. Baker, "Building, Visualizing, and Computing on Surfaces of Evolution," *IEEE Computer Graphics and Applications*, vol. 8, pp. 31–41, July 1988.
- [62] W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *ACM Computer Graphics*, vol. 21, pp. 163–169, July 1987.
- [63] H. E. Cline, C. L. Dumoulin, H. R. H. Jr., and W. E. Lorensen, "3D Reconstruction of the Brain from Magnetic Resonance Images Using a Connectivity Algorithm," *Magnetic Resonance Imaging*, vol. 5, pp. 345–352, July 1987.
- [64] H. E. Cline and W. E. Lorensen, "System and method for the display of surface structures contained within the interior region of a solid body." United States Patent Number 4,710,876, 1987.
- [65] M. J. Durst, "Additional Reference to "Marching Cubes"," *ACM Computer Graphics*, vol. 22, pp. 72–73, July 1988.
- [66] B. A. Payne and A. W. Toga, "Surface Mapping Brain Function on 3D Models," *IEEE Computer Graphics and Applications*, vol. 10, pp. 33–41, Sept. 1990.

- [67] J. K. Udupa, S. N. Srihari, and G. T. Herman, "Boundary Detection in Multidimensions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, pp. 41–50, Jan. 1988.
- [68] J. K. Udupa and G. T. Herman, "Boundaries in Multidimensional Digital Scenes: Theory and Algorithms," Technical Report MIPG182, Medical Image Processing Group, University Of Pennsylvania, Medical Image Processing Group, Department of Radiology, University Of Pennsylvania, Brockley Hall, Fourth Floor, 418 Service Drive, Philadelphia, Pennsylvania, 19104–6021, Jan. 1992.
- [69] J. K. Udupa, "Surface Connectedness in Digital Spaces: Theory and Algorithms," Technical Report MIPG188, Medical Image Processing Group, University Of Pennsylvania, Medical Image Processing Group, Department of Radiology, University Of Pennsylvania, Brockley Hall, Fourth Floor, 418 Service Drive, Philadelphia, Pennsylvania, 19104–6021, Oct. 1992.
- [70] M. Hall, "Defining surfaces from sampled data," in *Graphics Gems* (A. S. Glassner, ed.), ch. 10, pp. 552–557, Boston: Academic Press, Inc., 1990.
- [71] T. F. Banchoff, "Computer Graphics Tools for Rendering Algebraic Surfaces and for Geometry of Order," in *Geometric Analysis and Computer Graphics* (P. Concus, R. Finn, and D. A. Hoffman, eds.), vol. 17 of *Mathematical Sciences Research Institute Publications*, (New York, New York), pp. 31–37, Mathematical Sciences Research institute, Academic Press, May 1988.
- [72] J. Bloomenthal, "Polygonization of Implicit Surfaces," *Computer Aided Geometric Design*, vol. 5, pp. 341–355, 1988.
- [73] J. Bloomenthal, "Polygonization of Implicit Surfaces," Xerox PARC Technical Report 4, Xerox Corporation, Xerox Corporation, Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California, 94304, Dec. 1988.
- [74] M. Hall and J. Warren, "Adaptive Polygonalization of Implicitly Defined Surfaces," *IEEE Computer Graphics and Applications*, vol. 90, p. 6, Nov. 1990.
- [75] R. Ohbuchi, D. Chen, and H. Fuchs, "Incremental volume reconstruction and rendering for 3D ultrasound imaging," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, (Chapel Hill, North Carolina), pp. 312–323, The Society of Photo–Optical Instrumentation Engineers, SPIE, Oct. 1992.
- [76] U. Ganapathy and A. Kaufman, "3D acquisition and visualization of ultrasound data," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, (Chapel



- Hill, North Carolina), pp. 535–545, The Society of Photo-Optical Instrumentation Engineers, SPIE, Oct. 1992.
- [77] M. Davis, *The Undecidable : Basic Papers on Undecidable Propositions, Unsolvability Problems, and Computable Functions*. New York, New York: Raven Press, 1965.
  - [78] B. A. Payne and A. W. Toga, "Distance Field Manipulations of Surface Models," *IEEE Computer Graphics and Applications*, vol. 12, pp. 65–71, Jan. 1992.
  - [79] A. W. Toga, *Three-Dimensional Neuroimaging*. New York, New York: Raven Press, 1990.
  - [80] C. Frederick and E. L. Schwartz, "Brain Peeling: Viewing inside of a Laminate Three-Dimensional Solid," *The Visual Computer*, vol. 6, pp. 37–49, 1990.
  - [81] E. L. Schwartz, *Computational Neuroscience*. System Development Foundation Benchmark Series, Cambridge, Massachusetts: The MIT Press, 1990.
  - [82] E. L. Schwartz, "Computer-aided neuroanatomy of macaque visual cortex," in *Computational Neuroscience* (E. L. Schwartz, ed.), ch. 23, pp. 295–315, Cambridge, Massachusetts: The MIT Press, 1990.
  - [83] L. Kaufman, J. H. Kaufman, and J.-Z. Wang, "On cortical folds and neuromagnetic fields," *Electroencephalography and clinical Neurophysiology*, vol. 79, pp. 211–226, 1991.
  - [84] J.-Z. Wang, S. J. Williamson, and L. Kaufman, "Magnetic Source Images Determined by a Least-Field Analysis: The Unique Minimum-Norm Least-Squares Estimation," *IEEE Transactions on Biomedical Engineering*, vol. 39, pp. 665–675, July 1992.
  - [85] Z.-L. Lü and S. J. Williamson, "Spatial extent of coherent sensory-evoked cortical activity," *Experimental Brain Research*, vol. 84, pp. 411–416, 1991.
  - [86] S. J. Williamson, Z.-L. Lü, D. Karon, and L. Kaufman, "Advantages and Limitations of Magnetic Source Imaging," *Brain Topography*, vol. 4, no. 2, pp. 169–180, 1991.
  - [87] P. Frey and M. Gautherie, "Generation Automatique D'un Maillage 3D Dans Un Ensemble De Voxels: Application à la modélisation thermique numérique 3D en thérapie ultrasonore," *Innovation Et Technologie En Biologie Et Médecine*, vol. 12, pp. 428–442, Apr. 1991.



- [88] P. Frey and M. Gautherie, "Numerical Processing of 3D MRI Data prior to Thermal Modeling and Clinical Treatment Planning of Ultrasound Hyperthermia Treatment of Brain tumors," in *Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (J. P. Morucci, R. Plonsey, J. L. Coatrieux, and S. Laxminarayan, eds.), no. 14 in 5, (Paris), pp. 950–951, Engineering in Medicine and Biology Society, IEEE, Oct. 1992.
- [89] P. Frey, E. Engler, and M. Gautherie, "Numerical processing of data from three-dimensional magnetic resonance imaging prior to thermal modelling and clinical treatment planning," *Ultrasonics*, vol. 30, pp. 137–138, Feb. 1992.
- [90] P. K. Agarwal, *Intersection and Decomposition Algorithms for Planar Arrangements*. Cambridge: Cambridge University Press, 1991.
- [91] J. M. Montesinos, *Classical Tessellations and Three-Manifolds*. Universitex, Berlin: Springer Verlag, 1987.
- [92] J. Stillwell, *Geometry of Surfaces*. Universitex, Berlin: Springer Verlag, 1992.
- [93] C. D. Hansen and P. Hinker, "Massively Parallel Isosurface Extraction," in *Proceedings of Visualization '92* (A. E. Kaufman and G. M. Nielson, eds.), (Los Alamitos, California), pp. 133–139, IEEE Computer Society Technical Committee on Computer Graphics, IEEE Computer Society Press, Oct. 1992.
- [94] G. M. Nielson and B. Hamann, "The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes," in *Proceedings of Visualization 91* (G. M. Nielson and L. Rosenblum, eds.), vol. 2, pp. 83–91, 1991.
- [95] J. Wilhelms and A. V. Gelder, "Topological considerations in isosurface generation: Extended abstract," *ACM Computer Graphics*, vol. 25, pp. 79–86, Nov. 1990.
- [96] A. D. Kalvin, C. B. Cutting, B. Haddad, and M. E. Noz, "Constructing Topologically Connected Surfaces for the Comprehensive Analysis of 3D Medical Structures," in *SPIE Medical Imaging V: Image Processing*, vol. 1445, pp. 247–258, Society for Photo-Optical Instrumentation Engineers, 1991.
- [97] B. G. Baumgart, *Geometric Modeling for Computer Vision*. Ph.D. dissertation, Stanford University, Department of Computer Science, Aug. 1974.
- [98] B. Haddad, "A Geometric Representation for the Quantitative Analysis of 3D Medical Images," master's project, New York University, Computer Science Department, May 1991.

- [99] J. V. Miller, D. E. Breen, W. E. Lorensen, R. M. O'Bara, and M. J. Wozny, "Geometrically Deformed Models: A Method for Extracting Closed Geometric Models from Volume Data," *ACM Computer Graphics*, vol. 25, pp. 217–226, July 1991.
- [100] A. S. Glassner, "Computing Surface Normals for 3D Models," in *Graphics Gems* (A. S. Glassner, ed.), Graphics Gems, ch. 10, pp. 562–566, Boston: Academic Press, Inc., 1990.
- [101] A. S. Glassner, "Maintaining winged-edge models," in *Graphics Gems II* (J. Arvo, ed.), no. 2 in The Graphics Gems Series, ch. 4, pp. 191–201, Boston: Academic Press, Inc, 1991.
- [102] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," *ACM Computer Graphics*, vol. 26, pp. 65–70, July 1992.
- [103] M. Levoy, "A Taxonomy of Volume Visualization Algorithms." SIGGRAPH 91 Volume Visualization Course Notes, May 1991.
- [104] V. Argiro and W. Van Zandt, "Voxels: Data in 3-D," *Byte Magazine*, vol. 15, pp. 177–182, May 1992.
- [105] U. Tiede, K. H. Hoenhe, M. Bomans, A. Pommert, M. Riemer, and G. Wiebecke, "Surface Rendering: Investigation of Medical 3D-Rendering Algorithms," *IEEE Computer Graphics and Applications*, vol. 10, pp. 41–53, Mar. 1999.
- [106] J. K. Udupa, H.-M. Hung, and K.-S. Chuang, "Surface and Volume Rendering in Three-Dimensional Imaging: A Comparison," *Journal of Digital Imaging*, vol. 4, pp. 159–168, Aug. 1991.
- [107] A. Wallin, "Constructing Isosurfaces from CT Data," *IEEE Computer Graphics and Applications*, vol. 11, pp. 28–33, Nov. 1991.
- [108] B. Giles, "Contouring Data Fields," *Dr. Dobbs Journal*, vol. 25, pp. 44–46, June 1992.
- [109] P. J. Elliott, "Interactive image segmentation for radiation treatment planning," *IBM Systems Journal*, vol. 31, no. 4, pp. 620–634, 1992.
- [110] V. Klymenko and J. M. Coggins, "Visual Information Processing of Computed Topographic Electrical Activity Brain Maps," *Journal of Clinical Neurophysiology*, vol. 7, pp. 484–497, Apr. 1990.



- [111] D. Karron, J. Cox, and B. Mishra, "The Euler's Web proof that the Spider-Web Algorithm obeys the Euler-Poincaré Law," Robotics Laboratory Technical Report, New York University, Computer Science Department, 1992.
- [112] H. Nyquist, "Certain factors affecting telegraph speed," *Bell System Technical Journal*, vol. 3, pp. 324+, 1924.
- [113] H. Nyquist, "Certain topics in telegraph transmission theory," *American Institute of Electrical Engineers Transactions*, pp. 617+, Apr. 1928.
- [114] C. E. Shannon, "Communications in the presence of noise," *Proceedings of the Institute of Radio Engineers*, vol. 37, pp. 10–21, 1949.
- [115] J. Dugundji, *Topology*. Allyn and Bacon, 1966.
- [116] M. M. Lipschutz, *Differential Geometry*. Schaum's Outline Series, New York, New York: McGraw-Hill Book Company, 1969.
- [117] M. M. Lipschutz, *General Topology*. Schaum's Outline Series, New York, New York: McGraw-Hill Book Company, 1965.
- [118] B. Grünbaum and G. C. Shepard, *Tileings and Patterns : An Introduction*. New York, New York: W. H. Freeman and Company, 1987.
- [119] G. K. Francis, *A Topological Picturebook*. New York, New York: Springer-Verlag, 1987.
- [120] J. J. Koenderink, *Solid Shape*. The MIT Press Series in Artificial Intelligence, Cambridge, Massachusetts: The MIT Press, 1990.
- [121] A. H. Schoen, "Embedded Triply-Periodic Minimal Surfaces and Related Soap Film Experiments," in *Geometric Analysis and Computer Graphics* (P. Concus, R. Finn, and D. A. Hoffman, eds.), vol. 17 of *Mathematical Sciences Research Institute Publications*, (New York, New York), pp. 148–157, Mathematical Sciences Research institute, Academic Press, May 1988.
- [122] A. Kaufman, *Introduction to Volume Synthesis*, pp. 27–35. Tokyo: Springer-Verlay, 1991.
- [123] W.-J. Su, R. L. Woodward, and A. J. Dziewonski, "Deep origin of mid-ocean-ridge seismic velocity anomalies," *Nature*, vol. 360, pp. 149–152, Nov. 1992.
- [124] C. M. Hoffmann, *Geometric and Solid Modeling: An Introduction*. San Mateo, California, 94403: Morgan Kaufmann Publishers, Inc., 1989.



- [125] G. Chartrand and L. Lesniak, *Graphs and Digraphs*. The Wadsworth and Brooks/Cole Mathematics Series, Monterey, California: Wadsworth and Brooks/Cole Advanced Books and Software, second edition ed., 1986.
- [126] K. Akeley, "The Silicon Graphics 4D / 240GTX Superworkstations," *IEEE Computer Graphics and Applications*, vol. 9, pp. 71–83, July 1989.
- [127] V. Argiro, "Seeing in Volume," *Pixel*, vol. 5, pp. 35–39, July 1990.
- [128] J. Wilhelms and A. V. Gelder, "Octrees for faster isosurface generation," *ACM Transactions on Graphics*, vol. 11, pp. 201–227, July 1992.
- [129] J.-P. Thirion and A. Gourdon, "The 3D Marching Lines Algorithm and its Application to Crest Lines Extraction," *Rapports de Recherche 1672*, Institut National de Recherche en Informatique et en Automatique (INRA), Domane de Voluceau Rocquencourt, B. B. 105, 78153 Le Chesnay Cedex, France, May 1992.
- [130] E. Kreyszig, *Advanced Engineering Mathematics*. New York, New York: John Wiley and Sons, Inc., fifth ed., 1983.
- [131] J. M. Coggins, "Statistical investigations of multiscale image structure," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, (Chapel Hill, North Carolina), pp. 145–158, The Society of Photo-Optical Instrumentation Engineers, SPIE, Oct. 1992.
- [132] H. Schenker, *Five Graphic Music Analyses*. Dover reprint 1969, New York, New York: Dover, 1933.
- [133] H. Schenker, *Free Composition Der frier Satz*. Translation edition published 1979, New York, New York: Longmans, 1935.
- [134] H. W. Guggenheimer, *Differential Geometry*. Dover Books on Advanced Mathematics, New York, New York: Dover, dover corrected republication in 1977 of of mcgraw-hill book company publication ed., 1963.
- [135] D. Hilbert and S. Cohn-Vossen, *Geometry and the Imagination (Anschauliche Geometrie)*, ch. 4. New York, New York: Chelsea Publishing Company, 1932.
- [136] B. Mishra, "Notes on Solid Modeling," lecture notes, New York University, Computer Science Department, 1991.
- [137] G. Q. Maguire Jr., M. E. Noz, H. Rusinek, J. Jaeger, E. L. Kramer, J. J. Sanger, and G. Smith, "Graphics Applied to Medical Image Registration," *IEEE Computer Graphics and Applications*, vol. 11, no. 2, pp. 20–28, 1991.

- [138] D. L. Hykes, W. R. Hedrick, and D. E. Starchmann, *Ultrasound Physics and Instrumentation*. New York, New York: Churchill Livingstone, Inc., 1985.
- [139] M. Bajura, H. Fuchs, and R. Ohbuchi, "Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery within the Patient," *ACM Computer Graphics*, vol. 26, pp. 203–210, July 1992.
- [140] D. B. Karron, "The "SpiderWeb" algorithm for surface construction in noisy volume data," in *Visualization in Biomedical Computing '92* (R. A. Robb, ed.), vol. 1808, pp. 462–576, Society of Photo-Optical Instrumentation Engineers, 1992.
- [141] D. Karron, J. Cox, and B. Mishra, "The SpiderWeb Surface Construction Algorithm for Medical Imaging: Properties of Its Surface," Robotics Laboratory Technical Report, New York University, Computer Science Department, 1992.
- [142] D. Cohen, "Magnetoencephalography: Evidence of magnetic fields produced by alpha-rhythm currents," *Science*, vol. 161, pp. 784–786, 1968.
- [143] D. Cohen, E. A. Edelsack, and J. E. Zimmerman, "Magnetocardiograms taken inside a shielded room with a superconducting point-contact magnetometer," *Appl. Phys. Lett.*, vol. 16, pp. 278–280, 1970.
- [144] D. Brenner, S. J. Williamson, and L. Kaufman, "Visually evoked magnetic fields of the human brain," *Science*, vol. 190, pp. 480–482, 1975.
- [145] T. J. Teyler, B. N. Cuffin, and D. Cohen, "The visual evoked magnetoencephalogram," *Life Science*, vol. 17, pp. 683–692, 1975.
- [146] D. Brenner, J. Lipton, L. Kaufman, and S. J. Williamson, "Somatically evoked magnetic fields of the human brain," *Science*, vol. 199, pp. 81–83, 1978.
- [147] G. M. Baule and R. McFee, "Detection of the magnetic field of the heart," *Am. Heart J.*, vol. 66, pp. 95–96, 1963.
- [148] T. Yamamoto, S. J. Williamson, L. Kaufman, C. Nicholson, and R. Llinás, "Magnetic Localization of Neuronal Activity in the Human Brain," *Proceedings of the National Academy of Science, USA*, vol. 85, pp. 8732–8736, 1988.
- [149] G. L. Romani, S. J. Williamson, and L. Kaufman, "Tonotopic organization of the human auditory cortex," *Science*, vol. 216, pp. 1339–1340, 1982.
- [150] S. J. Williamson and L. Kaufman, "Biomagnetism," *J. Magn. Magn. Mat.*, vol. 22, pp. 129–201, 1981.



- [151] L. Kaufman and S. J. Williamson, "Neuromagnetic studies of sensory functions and mental imagery," p. in press, 1991.
- [152] A. I. Ahonen, M. S. Hämäläinen, M. J. Kajola, J. E. T. Knuutila, O. V. Lounasmaa, J. T. Simola, C. D. Tesche, and V. A. Vilkmán, "Multichannel SQUID Systems for Brain Research," *IEEE Transactions on Magnetics*, vol. MAG-27, pp. 2786–2792, 1991.
- [153] A. I. Ahonen, M. S. Hämäläinen, M. J. Kajola, J. E. T. Knuutila, P. P. Laine, O. V. Lounasmaa, J. T. Simola, C. D. Tesche, and V. A. Vilkmán, "A 122-Channel Magnetometer Covering the Whole Head," in *Symposium on Neuroscience and Technology*, p. in press, IEEE Engineering in Medicine and Biology, 1992.
- [154] Biomagnetic Technology Inc, "BTI MAGNES Neuromagnetometer Users Manual," technical manual, Biomagnetic Technologies, Inc., San Diego, California, 1985.
- [155] R. Plonsey, *Bioelectric Phenomena*. New York, New York: McGraw-Hill Book Company, 1969.
- [156] S. J. Williamson and L. Kaufman, "Analysis of Neuromagnetic Signals," in *Methods of Analysis of Brain Electrical and Magnetic Signals* (A. S. Gevins and A. Rémond, eds.), vol. 1 of *Handbook of Electroencephalography and Clinical Neurophysiology, Revised Series*, ch. 14, pp. 405–448, Amsterdam: Elsevier, 1987.
- [157] S. J. Williamson and L. Kaufman, "Theory of neuroelectric and neuromagnetic fields," in *Auditory Evoked Magnetic Fields and Electric Potentials* (F. Grandori, H. Hoke, and G. L. Romani, eds.), vol. 6 of *Advances in Audiology*, pp. 1–39, Basel: Karger, 1990.
- [158] K. R. Swinney and J. P. Wikswo, "A calculation of the magnetic field of a nerve action potential," *Biophys. J.*, vol. 32, pp. 719–731, 1980.
- [159] S. Webb, *From the Watching of Shadows: The Origins of Radiological Tomography*. Bristol, United Kingdom: Institute of Physics Publishing Inc., 1992.
- [160] I. I. Rabi, S. Millman, and P. Kusch, "The molecular beam resonance method for measuring nuclear magnetic moments," *Physical Review*, vol. 55, p. 526, 1939.
- [161] F. Bloch, W. W. Hansen, and M. Packard, "Nuclear induction," *Physical Review*, vol. 69, pp. 127+, 1946.



- [162] E. M. Purcell, H. C. Torrey, and R. V. Pound, "Resonance absorption by nuclear magnetic moments in a solid," *Physical Review*, vol. 69, pp. 37+, 1946.
- [163] P. C. Lauterbur, "Image formation by induced local interactions: examples employing nuclear magnetic resonance," *Nature*, vol. 242, pp. 190–191, 1973.
- [164] J. R. Mallard, "Magnetic Resonance Imaging – The odyssey of one contributor to its birth," in *Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (J. P. Morucci, R. Plonsey, J. L. Coatrieux, and S. Laxminarayan, eds.), no. 14, pp. 2860–2862, Oct. 1992.
- [165] R. Damadian, "Tumor detection by NMR," *Science*, vol. 171, pp. 1151–1153, 1971.
- [166] R. Damadian, M. Goldsmith, and L. Minkoff, "Fonar image of the live human body," *Physiological Chemistry and Physical Medicine NMR*, vol. 97, pp. 97+, 1977.
- [167] R. R. Edelman, J. Kleefield, K. U. Wentz, and D. J. Atkinson, *Basic Principles of Magnetic Resonance Imaging*, pp. 3–38. Philadelphia: W. B. Saunders, 1990.
- [168] Phillips Medical Systems, "Gyrosan 1.5 operators manual," technical manual, Phillips Medical Systems, 1985.
- [169] M. R. Spiegel, *Schaum's Outline of Theory and Problems of Advanced Mathematics for Engineers and Scientists*, ch. 5. Schaum's Outlines, New York, New York: McGraw-Hill Book Company, 1971.
- [170] E. Kreyszig, *Advanced Engineering Mathematics*, ch. 6. New York, New York: John Wiley and Sons, Inc., fifth ed., 1983.
- [171] P. C. Shields, *Elementary Linear Algebra*, ch. 4, p. 268. New York, New York: Worth Publishers, Inc., 1980.
- [172] F. L. Bookstein, "Part 3, Analytic Methods: Introduction and Overview: Geometry and Biology," in *Proceedings of the Michigan Morphometrics Workshop* (F. J. Rohlf and F. L. Bookstein, eds.), vol. 2, pp. 61–74, Ann Arbor: Museum of Zoology, The University of Michigan, 1990.
- [173] J-P Thirion, A. Gourdon, O. Monga, A. Gueziec, and N. Ayache, "Fully Automatic Registration of 3d CAT-Scan Images Using Crest Lines," in *Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine*

- and Biology Society* (J. P. Morucci, R. Plonsey, J. L. Coatrieux, and S. Laxminarayan, eds.), no. 14 in 5, (Paris), pp. 1888–1890, Engineering in Medicine and Biology Society, IEEE, Oct. 1992.
- [174] A. Guéziec and N. Ayache, "Smoothing and Matching of 3-D Space Curves," *Rapports de Recherche 1544*, Institut National de Recherche en Informatique et en Automatique (INRA), Domane de Voluceau Rocquencourt, B. B. 105, 78153 Le Chesnay Cedex, France, Oct. 1991.
  - [175] S. J. Williamson, M. Hoke, G. Stroink, and M. Kotani, eds., *Advances in Biomagnetism*. New York, New York: Plenum, 1989.
  - [176] H. H. Jasper, "The Ten-Twenty Electrode System of the International Federation of Electroencephalography and Clinical Neurophysiology," *Electroencephalography Clinical Neurophysiology*, vol. 10, pp. 371–375, 1958.
  - [177] Biomagnetic Technology Inc, "BTI Gemini Neuromagnetometer Users Manual," technical manual, Biomagnetic Technologies, Inc., San Diego, California, 1985.
  - [178] S. J. Williamson, "Neuromagnetometer Laboratory Manual," technical manual, New York University, 1985.
  - [179] W. E. Boyes, *Jigs and Fixtures*. Society of Mechanical Engineers, Dearborn, Mich., 1982.
  - [180] F. B. Hazen and P. Wright, "Workholding Automation: Innovations in Analysis, Design and Planning," *Manufacturing Review*, vol. 3, pp. 224–237, Dec. 1990.
  - [181] B. Mishra, "Workholding—Analysis and Planning," in *Proceedings: IEEE/RSJ International Workshop on Intelligent Robots and Systems: IROS '91*, vol. 1, (Osaka, Japan), pp. 53–57, Nov. 1991.
  - [182] P. Ferreira, B. Kochar, C. Liu, and V. Chandru, "AFIX: An Expert System Approach to Fixture Design," in *ASME Winter Annual Meeting*, pp. 73–82, Computer-Aided/Intelligent Process Planning, Nov. 1985.
  - [183] H. Asada and A. By, "Kinematic Analysis of Workpart Fixturing for Flexible Assembly with Automatically Reconfigurable Fixtures," *IEEE Journal of Robotics and Automation*, vol. RA-1, pp. 86–94, Dec. 1985.
  - [184] A. B. Abche, G. S. Tzanakos, and E. Micheli-Tzanakou, "A Monte-Carlo Evaluation of Multimodal 3-D Image Registration Using External Markers," in *Proceedings of the 14th Annual International Conference of the IEEE Engineering*



- in Medicine and Biology Society* (J. P. Morucci, R. Plonsey, J. L. Coatrieux, and S. Laxminarayan, eds.), no. 14, pp. 1885–1887, Oct. 1992.
- [185] F. L. Bookstein, *Morphometric Tools for Landmark Data*. Cambridge: Cambridge University Press, 1991.
  - [186] F. L. Bookstein, *Introduction to Methods for Landmark Data*, ch. 3, pp. 215–225. Proceedings of the Michigan Morphometrics Workshop, Ann Arbor, Michigan: University of Michigan Museum of Zoology, May 1990.
  - [187] Y. L. Kergosien, “Generic Sign Systems in Medical Imaging,” *IEEE Computer Graphics and Applications*, vol. 11, pp. 46–65, July 1991.
  - [188] C. B. Cutting, J. G. McCarthy, and D. B. Karron, “Three-Dimensional Input of Body Surface Data Using a Laser Light Scanner,” *Annals of Plastic Surgery*, vol. 21, pp. 38–45, July 1988.
  - [189] Cyberware Laboratories, Inc, “Cyberware<sup>TM</sup> Light Scanner User’s Manual,” technical users manual, Cyberware Laboratories, Inc., 8 Harris Court, Suite 3D, Monterey, Ca., 93940, 1992.
  - [190] M. W. Vannier, T. Pilgram, G. Bhatia, B. Brunsten, and P. Commean, “Facial Surface Scanner,” *IEEE Computer Graphics and Applications*, vol. 11, pp. 71–80, Nov. 1991.
  - [191] G. Bhatia, A. Godhwani, and J. R. Grindon, “Optimal 3D Surface Metrology—Localization of Fiducial Points,” *IEEE Proceedings Southeastcon91*, vol. 2, pp. 925–930, 1991.
  - [192] B. S. Savara, S. H. Miller, R. J. Demuth, and H. K. Kawamoto, “Biostereometrics and Computer Graphics for Patients with Craniofacial Malformations: Diagnosis and Treatment Planning,” *Plastic and Reconstructive Surgery*, vol. 75, pp. 495–449, Apr. 1985.
  - [193] L. G. Farkas, W. Bryson, and J. Klotz, “Is photogrammetry of the face reliable?,” *Plastic and Reconstructive Surgery*, vol. 66, pp. 346–355, Sept. 1980.
  - [194] A. Guéziec and N. Ayache, “Smoothing and Matching of 3-D Space Curves,” in *Visualization in Biomedical Computing* (R. Robb, ed.), no. 1808, (Chapel Hill, North Carolina), pp. 259–273, The Society of Photo-Optical Instrumentation Engineers, SPIE, Oct. 1992.



- [195] O. Monga, S. Benayoun, and O. D. Faugeras, "From Partial Derivatives of 3D Density Images to Ridge Lines," in *Visualization in Biomedical Computing* (R. Robb, ed.), no. 1808, (Chapel Hill, North Carolina), pp. 118–129, The Society of Photo-Optical Instrumentation Engineers, SPIE, Oct. 1992.
- [196] B. Grayson, C. Cutting, F. L. Bookstein, H. Kim, and J. G. McCarthy, "Three-Dimensional Cephalogram: Theory, Technique, and Clinical Application," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 94, pp. 327–337, Oct. 1988.
- [197] S. J. Chaconas, G. A. Engel, A. A. Gianelly, J. C. Gorman, D. C. Grummonds, M. S. Lemchen, and R. S. Nanda, "The DigiGraph Work Station: Part 1, Basic Concepts," *Journal of Clinical Orthodontics*, vol. 24, pp. 360–367, June 1990.
- [198] R. G. Alexander, J. C. Gorman, D. C. Grummonds, R. L. Jacobson, and M. S. Lemchen, "The DigiGraph Work Station: Part 2, Clinical Management," *Journal of Clinical Orthodontics*, vol. 24, pp. 402–407, July 1990.
- [199] S. J. Chaconas, R. L. Jacobson, and M. S. Lemchen, "The DigiGraph Work Station: Part 3, Accuracy of Cephalometric Analysis," *Journal of Clinical Orthodontics*, vol. 24, pp. 467–471, Aug. 1990.
- [200] R. Bruyer, *The Neuropsychology of face perception and facial expression*. Hillside, New Jersey: Lawrence Erlbaum Associates, 1986.
- [201] T. J. Sejnowski and P. S. Churchland, *Brain and Cognition*. Cambridge, Massachusetts: The MIT Press, 1989.
- [202] H. Pohjonen, A. Kiuru, P. Nikkinen, P. Kark, J. Ylä-Jääski, and T. Vehkomäki, "Registration of Anatomical and Functional 3D-Data Sets," in *Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (J. P. Morucci, R. Plonsey, J. L. Coatrieux, and S. Laxminarayan, eds.), no. 14, pp. 2239–2240, Oct. 1992.
- [203] C. A. Pelizzari, G. T. Y. Chen, D. R. Spelbring, R. R. Weichselbaum, and C.-T. Chen, "Accurate Three-Dimensional Registration of CT, PET, and/or MR Images of the Brain," *Journal of Computer Assisted Tomography*, vol. 13, no. 1, pp. 20–26, 1989.
- [204] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*. Cambridge: Cambridge University Press, 1988.

- [205] A. C. Evans, S. Marrett, J. Torrescorzo, S. Ku, and L. Collins, "MRI-PET Correlation in Three Dimensions Using a Volume-Of-Interest (VOI) Atlas," *Journal of Cerebral Blood Flow and Metabolism*, vol. 11, pp. A69-A78, 1991.
- [206] P. Schonemann, "A generalized Solution of the Orthogonal Procrustes Statistics," *Journal of the Statistical Society B*, vol. 40, pp. 234-238, 1978.
- [207] F. J. Rohlf and D. Slice, "Extensions of the Procrustes Method for the Optimal Superimposition of Landmarks," *Systematic Zoology*, vol. 39, no. 1, pp. 40-59, 1990.
- [208] F. J. Rohlf, *Rotational Fit (Procrustes) Methods*, ch. 3, pp. 215-225. Proceedings of the Michigan Morphometrics Workshop, Ann Arbor, Michigan: University of Michigan Museum of Zoology, second ed., May 1990.
- [209] M. E. Noz and G. Q. M. Jr., "QSH: A Minimally but Highly Portable Image Display and Processing Toolkit," *Computer Methods and Programs in Biomedicine*, vol. 27, no. 11, pp. 229-240, 1988.
- [210] G. Q. Maguire Jr., M. E. Noz, E. M. Lee, and J. H. Schimpf, "Correlation Methods for Tomographic Images Using Two and Three Dimensional Techniques," in *Information Processing in Medical Imaging* (S. L. Bacharach, ed.), pp. 266-279, Dordrecht: Martinus Nijhoff Publishers, 1985.
- [211] T. G. Turkington, R. J. Jaszczak, K. L. Greer, R. E. Coleman, and C. A. Pelizzari, "Correlation of SPECT images of a Three-dimensional Brain Phantom Using a Surface Fitting Technique," *IEEE Transactions on Nuclear Science*, vol. 39, pp. 1460-1463, Oct. 1992.
- [212] G. Q. Maguire Jr., J. Jaeger, L. Farde, and M. E. Noz, "Use of Graphical Techniques for Error Evaluation," *Journal of Medical Systems*, vol. 11, no. 4, pp. 277-286, 1987.
- [213] M. E. Brummer, "Hough Transform Detection of the Longitudinal Fissure In Tomographic Head Images," *IEEE Transactions on Medical Imaging*, vol. 10/1, pp. 74-81, 1991.
- [214] P. V. C. Hough, "Method and Means for Recognizing Complex Patterns." United States Patent number 3,069,654, 1962.
- [215] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11-15, 1972.



- [216] Q.-S. Chen, M. Defrise, F. Deconinck, and J. H. Jonickheer, "3D Multimodality Imaging: The Registration Problem," in *Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (J. P. Morucci, R. Plonsey, J. L. Coatrieux, and S. Laxminarayan, eds.), no. 14, pp. 2168–2169, Oct. 1992.
- [217] P. T. Fox, J. S. Perlmutter, and M. E. Raichle, "A Stereotactic Method of Anatomical Localization for Positron Emission Tomography," *Journal of Computer Assisted Tomography*, vol. 9, pp. 597–610, 1984.
- [218] R. A. Robb, *Three-Dimensional Biomedical Imaging, Volume 1*. Boca Raton, Florida: CRC Press, Inc., 1985.
- [219] R. A. Robb, *Three-Dimensional Biomedical Imaging, Volume 2*. Boca Raton, Florida: CRC Press, Inc., 1985.
- [220] R. A. Robb, D. P. Hanson, R. A. Karwoski, A. G. Larson, E. L. Workman, and M. C. Stacy, "ANALYZE: A comprehensive, operator-interactive software package for multidimensional medical image display and analysis," *Computerized Medical Imaging and Graphics*, vol. 16, no. 6, pp. 433–454, 1989.
- [221] R. A. Robb and C. Barillot, "Interactive Display and Analysis of 3-D Medical Images," *IEEE Transactions on Medical Imaging*, vol. 8, pp. 217–226, Sept. 1989.
- [222] R. A. Robb and D. P. Hanson, "ANALYZE: A software system for biomedical image analysis," in *Proceedings of the first conference on visualization in biomedical computing*, (Los Alamitos, CA), pp. 507–518, IEEE Computer Society, IEEE Computer Society Press, May 1990.
- [223] R. A. Robb and D. P. Hanson, "A software system for interactive and quantitative visualization of multidimensional biomedical images," *Australasian Physical and Engineering Sciences in Medicine*, vol. 14, no. 1, pp. 9–30, 1991.
- [224] H. Jiang, R. A. Robb, and K. S. Holton, "A New Approach to 3-D Registration of Multimodality Medical Images by Surface Matching," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, (Chapel Hill, North Carolina), pp. 196–213, The Society of Photo-Optical Instrumentation Engineers, SPIE, Oct. 1992.
- [225] E. M. Friets, J. W. Strohbehn, J. F. Hatch, and D. W. Roberts, "A frameless stereotaxic operating microscope for neurosurgery," *IEEE Transactions on Biomedical Engineering*, vol. 36, pp. 608–617, 1989.



- [226] H. J. Wieringa and M. J. Peters, "MRI and MEG, 3-D Models and Display," *In Press*, vol. 0, p. 0, Sept. 1991.
- [227] D. L. King, D. L. King, Jr, and M. Y. Shao, "Three Dimensional Spatial Registration and Interactive Display of Position and Orientation of Real-Time Ultrasound Images," *Journal of Ultrasound Medicine*, vol. 9, pp. 252-532, 1990.
- [228] D. Regan, *Human Brain Electrophysiology*. New York, New York: Elsevier, 1989.
- [229] D. H. Enlow, *Handbook of facial growth*. Philadelphia, Pennsylvania: W. B. Saunders, 1975.
- [230] F. J. Rohlf and F. L. Bookstein, *Proceedings of the Michigan Morphometrics Workshop*, vol. 2 of *Special Publication*. Ann Arbor, Michigan: University of Michigan Museum of Zoology, second ed., May 1990.
- [231] E. Hamilton, *Mythology*, ch. 11, pp. 210-211. Boston, Massachusetts: Little, Brown and Company, second ed., 1942.
- [232] F. L. Bookstein and W. D. K. Green, "Edge information and landmarks in medical images," in *Visualization in Biomedical Computing* (R. Robb, ed.), 1808, (Chapel Hill, North Carolina), pp. 242-258, The Society of Photo-Optical Instrumentation Engineers, SPIE, Oct. 1992.
- [233] S. J. Williamson and L. Kaufman, "Theory of Neuroelectric and Neuromagnetic Fields," in *Auditory Electric and Magnetic Fields* (F. Grandori, H. Hoke, and G. L. Romani, eds.), vol. 6, pp. 1-39, Basel: Karger, 1989.
- [234] B. Pansky, *Review of Gross Anatomy*. New York, New York: Macmillan Publishing, Inc., fourth ed., 1979.
- [235] C. D. Clemente, *Anatomy: A Regional Atlas of the Human Body*. Baltimore: Urban and Schwarzenberg, 1981.
- [236] H. Goldstein, *Classical mechanics*. Addison-Wesley Series in Physics, Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., second edition ed., 1980.
- [237] K. R. Symon, *Mechanics*. Addison-Wesley Series in Physics, Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., third edition ed., 1971.
- [238] G. Strang, *Linear Algebra and its Applications, Third Edition*. San Diego: Harcourt Brace Jovanovich, 1988.

- [239] M. M. Lipschutz, *Linear Algebra*. Schaum's Outline Series, New York, New York: McGraw-Hill Book Company, 1968.
- [240] S. J. Williamson and L. Kaufman, "Advances in Neuromagnetic Instrumentation and Studies of Spontaneous Brain Activity," *Brain Topography*, vol. 2, pp. 129–139, 1989.
- [241] J. Alakuijala, J. Oikarinen, Y. Louhisalmi, X. Ying, and J. Joivukangas, "Image Transformation from Polar to Cartesian Coordinates Simplifies the Segmentation of Brain Images," in *Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (J. P. Morucci, R. Plonsey, J. L. Coatrieux, and S. Laxminarayan, eds.), no. 14, pp. 1918–1919, Oct. 1992.
- [242] G. H. Golub and C. F. V. Loan, *Matrix Computations, Second Edition*. Baltimore: The Johns Hopkins University Press, 1989.
- [243] F. H. Raab, E. B. Blood, T. O. Steiner, and H. R. Jones, "Magnetic Position and Orientation Tracking System," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 15, pp. 709–718, Sept. 1979.
- [244] R. Hari and R. J. Ilmoniemi, "Cerebral magnetic fields," *CRC Critical Reviews in Biomedical Engineering*, vol. 14, pp. 93–126, 1986.
- [245] A. C. Bruno and S. J. Williamson, "Representations of Isocontours on a Spherical Head," in *Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (J. P. Morucci, R. Plonsey, J. L. Coatrieux, and S. Laxminarayan, eds.), vol. 14, (Piscataway, NJ), pp. 2174–2178, IEEE Engineering in Medicine and Biology Society, 1992.
- [246] F. Aurenhammer, "Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys*, vol. 12, pp. 345–406, Sept. 1991.
- [247] J. D. Jackson, *Classical Electrodynamics*. New York, New York: John Wiley and Sons, Inc, 1962.
- [248] R. Plonsey and R. C. Barr, *Bioelectricity: A Quantitative Approach*. New York, New York: Plenum Press, 1988.
- [249] H. Helmholtz, "Über einige Gesetze der Vertheilung elektrischer Ströme in körperlichen Leitern, mit Anhwendung auf die thierisch–elektrischen Versuche," *Annals of Physical Chemistry*, vol. 89, pp. 211–233, 353–377, 1853.



- [250] C. W. Steele, *Numerical Computation of Electric and Magnetic Fields*. New York, New York: Van Nostrand Reinhold Company Inc., 1987.
- [251] Z.-L. Lü, *Neuromagnetic Investigations of Spontaneous and Sensory-Evoked Activity of Human Cerebral Cortex*. Ph.D. dissertation, New York University, Department of Physics, Mar. 1992.
- [252] M. S. Hämäläinen and R. J. Ilmoniemi, "Interpreting measured magnetic fields of the brain: Estimates of current distributions," Tech. Rep. TKK-F-A559, Helsinki University of Technology, 1984.
- [253] M. S. Hämäläinen and R. J. Ilmoniemi, "Interpreting magnetic fields of the brain: Minimum norm estimates," *IEEE Trans. Biomed. Engng.*, p. in press, 1991.
- [254] R. Hari, K. Aittoniemi, M.-L. Järvinen, T. Katila, and T. Varpula, "Auditory evoked transient and sustained magnetic fields of the human brain: Localization of neural generators," *Exp. Brain Res.*, vol. 40, pp. 237–240, 1980.
- [255] J. P. W. Jr. and J. P. Barach, "An estimate of the steady magnetic field strength required to influence nerve conduction," *IEEE Trans. Biomed. Eng.*, vol. BME-27, pp. 722–723, 1980.
- [256] R. Plonsey, "Magnetic field resulting from action currents on cylindrical fibers," *Med. Biol. Eng. and Comp.*, vol. 19, pp. 311–315, 1981.
- [257] B. J. Roth and J. P. Wikswo, Jr., "The magnetic field of a single axon," *Biophys. J.*, vol. 48, pp. 93–109, 1985.
- [258] H. Berger, "Über das Elektroenkephalogramm des Menschen," *Archiv. für Psychiat. und Nervenkr.*, vol. 87, pp. 527–570, 1929.
- [259] H. Berger, "Über das Elektroenkephalogramm des Menschen. II.," *Journal of Psychological Neurology (Lpz.)*, vol. 40, pp. 160–179, 1930.
- [260] P. Andersen and S. A. Andersson, *Physiological Basis of the Alpha Rhythm*. New York, New York: Appleton-Century-Crofts, 1968.
- [261] M. Steriade, P. Gloor, R. R. Llinás, F. H. Lopes da Silva, and M. Mesulam, "Basic mechanisms of cerebral rhythmic activities," *Electroenceph. Clin. Neurophysiol.*, vol. 76, pp. 481–508, 1990.
- [262] F. Lopes da Silva, "Neural Mechanisms Underlying Brain Waves: From Neural Membranes to Networks," *Electroencephalography and Clinical Neurophysiology*, vol. 79, pp. 81–93, 1991.



- [263] G. E. Chatrian, M. C. Petersen, and J. A. Lazarte, "The blocking of the rolandic wicket rhythm and some central changes related to movement," *Electroenceph. clin. Neurophysiol.*, vol. 11, pp. 497–510, 1959.
- [264] J. Tiihonen, R. Hari, M. Kajola, J. Karhu, S. Ahlfors, and S. Tissari, "Magnetoencephalographic 10-hz rhythm from the human auditory cortex," *Neurosci. Lett.*, p. in press, 1991.
- [265] L. Kaufman, J.-Z. Wang, and S. J. Williamson, "Identifying Cortical Sources of Incoherent Fields of the Human Brain," in *Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (J. P. Morucci, R. Plonsey, J. L. Coatrieux, and S. Laxminarayan, eds.), vol. 14, (Piscataway, NJ), pp. 1760–1762, IEEE Engineering in Medicine and Biology Society, 1992.
- [266] E. R. John, L. S. Pritchep, J. Fridman, and P. Easton, "Neurometrics: Computer-assisted differential diagnosis of brain dysfunctions," *Science*, vol. 239, pp. 162–169, 1988.
- [267] G. Pfurtscheller and A. Aranibar, "Event-related Cortical Desynchronization Detected by Power Measurements of Scalp EEG," *Electroenceph. clin. Neurophysiol.*, vol. 42, pp. 817–826, 1977.
- [268] G. Pfurtscheller, D. Flotzinger, W. Mohl, and M. Peltoranta, "Prediction of the Side of Hand Movements from Single-trial Multi-channel EEG Data Using Neural Networks," *Electroenceph. Clin. Neurophysiol.*, vol. 82, pp. 313–315, 1992.
- [269] L. Kaufman, B. Schwartz, C. Salustri, and S. J. Williamson, "Modulation of spontaneous brain activity during mental imagery," *J. Cognitive Neuroscience*, vol. 2, pp. 124–132, 1990.
- [270] L. Kaufman, Y. Cykowicz, M. Glanzer, and S. J. Williamson, "Selective suppression of spontaneous cortical rhythms for cognitive tasks of imaging and rhyming," *J. Cognitive Neurosci.*, p. submitted, 1991.
- [271] F. H. L. da Silva and W. S. van Leeuwen, "The Cortical Alpha Rhythm in Dog: The Depth and Surface Profile of Phase," in *Architectonics of the Cerebral Cortex* (M. A. B. Brazier and H. Petsche, eds.), pp. 319–333, New York, New York: Raven Press, 1978.
- [272] F. H. L. da Silva and W. S. van Leeuwen, "The Cortical Source of the Alpha Rhythm," *Neurosci. Lett.*, vol. 6, pp. 237–241, 1977.

- [273] P. L. Nunez, "Generation of Human EEG by a Combination of Long and Short Range Neocortical Interactions," *Brain Topography*, vol. 1, pp. 199–215, 1989.
- [274] A. van Rotterdam, F. H. L. da Silva, J. van den Ende, M. A. Viergever, and A. J. Hermans, "A model of the spatial-temporal characteristics of the alpha rhythm," *Bull. Math. Biol.*, vol. 44, pp. 283–305, 1982.
- [275] T. H. Bullock and M. C. McClune, "Lateral coherence of the electroencephalogram: A new measure of brain synchrony," *Electroenceph. clin. Neurophysiol.*, vol. 73, pp. 479–498, 1989.
- [276] R. M. Chapman, R. J. Ilmoniemi, S. Barbanera, and G. L. Romani, "Selective localization of alpha brain activity with neuromagnetic measurements," *Electroenceph. Clin. Neurophysiol.*, vol. 58, pp. 569–572, 1984.
- [277] V. L. Vvedensky, S. P. Naurzakov, V. I. Ozhogin, and S. Y. Shabanov, "Measurement of the Tangential Component of the Magnetic Field Associated with Rhythmic Alpha Activity in the Human Brain," in *Fifth World Conference on Biomagnetism* (H. Weinberg, G. Stroink, and T. Katila, eds.), (New York, New York), pp. 57–60, Pergamon Press, 1985.
- [278] R. J. Ilmoniemi, S. J. Williamson, and W. E. Hostetler, "New method for the study of spontaneous brain activity," in *Biomagnetism '87* (K. Atsumi, M. Kotani, S. Ueno, T. Katila, and S. J. Williamson, eds.), (Tokyo), pp. 182–185, Tokyo Denki University Press, 1988.
- [279] S. J. Williamson, J.-Z. Wang, and R. J. Ilmoniemi, "Method for Locating Sources of Human Alpha Activity," in *Advances in Biomagnetism* (S. J. Williamson, M. Hoke, G. Stroink, and M. Kotani, eds.), (New York, New York), pp. 257–260, Plenum, 1989.
- [280] D. E. Dick and A. O. Vaughn, "Mathematical Description and Computer Detection of Alpha Waves," *Math. Biosci.*, vol. 7, pp. 81–95, 1970.
- [281] R. Kalimi, "Statistical properties of human alpha rhythm," Master's thesis, New York University, 1991.
- [282] P. C. Ribeiro, S. J. Williamson, and L. Kaufman, "SQUID Arrays for Simultaneous Magnetic Measurements: Calibration and Source Localization Performance," *IEEE Transactions on Biomedical Engineering*, vol. 35, pp. 551–560, 1988.

- [283] S. J. Williamson and L. Kaufman, "Magnetic fields of the cerebral cortex," in *Biomagnetism* (S. N. Erne, H. Hahlbohm, and H. Lübbig, eds.), pp. 353–402, Berlin: Walter de Gruyter, 1981.



# Index

- $\tau$ , 40, 135, 136, 157
  - defined, 137
- 2 hit faces, 136
- 2–4 Voxel face hit rule, ix
- 26-connectivity, 27
- 4 hit face, 145
- 4-hits, 150
- 6-connectivity, 27
- accurate, 252
- adaptive algorithms, 29
- adaptive polygonalization, 29
- adjacent voxel complementary face, 87
- adjacent voxels
  - figure, 80
- affine hull, 162
- alias transform, 186
  - defined, 186
- ALLIGATOR, 27, 31
- alpha rhythm, viii, 315–317, 319
- alphon, 3, 204, 318, 320, 328–330
  - defined, 316, 317
- Amoeba algorithm, 213
- analytic surface, 136
- ANALYZE, 204
- anterior tragus, 209, 210
- Anterior–Posterior Axis, 214
  - defined, 187
- array
  - defined, 77
- articulating point, 75, 151, 158, 160, 164
  - defined, 89
- ascending vertex values, 137
- auditory meatus, 216
- axial
  - defined, 212
- axial orientation
  - defined, 212
- balloon surface, 31
- bas relief, 13, 63
  - figure, 15
- basis directions, 131
- best line and point algorithm, 198
- Best Point and Line, 3
- best possible solution, 241
- BestLine subroutine, 230
- bet, 75
- bevel voxel edges, 27
- bifurcation, 152, 154, 155, 162, 163
- bifurcation edge, 159
- bilinear interpolation, 30, 34, 58, 89
- Biomagnetic Technology, Inc., 189
- birth voxels, 140
- blobby trace centers, 240
- blobby traces, 244
- Blobulator, 225, 230
  - figure, 226, 227
- BlocknessFactor
  - figure, 120
- boundary element analysis, 23, 29
- boundary surface, 26
- bounded
  - defined, 154
- boundedness, 77

- defined for 2D, 40
  - defined for 3D, 76
- boxel, 35
  - defined, 35, 40
- boxel vertex
  - defined, 40
- boxels, 81
- brain
  - biomagnetic fields, 175
- bubble, 9, 48, 75
- bubble resolution phase, 163
- bubble triangles, 151, 162
- bubble voxel face, 151
- bubble voxels, 116
- bubbles, 71, 127, 150, 153
  - in contour rings, 36
  - popped, 161
- bug
  - defined, 246
  - surface cursor, 246
- canonical directions, 240
- cardinal landmark, 173
- cardinal landmarks, 149, 173, 174, 187–190, 194, 195, 198, 202, 204, 206, 209, 210, 212, 213, 217, 222–224, 228, 230, 238, 245, 246, 249, 251–253, 256, 257, 263, 278, 283, 284, 318, 390
  - defined, 188
- cardinal landmarks, 187
- cardiomagnetic fields, 175
- case table, 87, 103
- case tables, 66
- cases, 89
- Cauchy's theorem, 136
- CEMAX Medical Systems, 24
- center of gravity, 227
- centroid, 58, 89
- cephalogram, 197
- cerebral spinal fluid, 73
- cerebro-spinal fluid, 202
- certainty, 153
- chamfer matching, 204
- ChapVolumes, 27
- chirality, 187
- circulation
  - defined, 86
  - of simplicial mesh, 93
- clean segmentation, 73
- closed
  - defined, 154
- closed path, 43
- closed surface
  - defined, 156
- CM-2, 30
- co-planar, 118
- co-planar triangles, 31, 75, 97
- cobweb, 89
- coherently orientable mesh, 93
- coherently oriented
  - defined, 156
- commutative property, 93
- compact, 153
  - defined, 155
- compact two manifold
  - defined, 155
- complementary voxel faces
  - defined, 86
- complex
  - defined, 93
- complex analysis, 136
- Comprehensive Source Searching, 308
- computer aided surgery, 25
- confidence, 69
- congruent, 85, 86
  - defined, 86
- congruent coordinate system, 217
- connected edges, 152

- defined, 85
- connected faces, 152
  - defined, 81
  - figure, 83
- connected hits, 89, 152
  - defined, 85
  - figure, 84
- Connection Machine, 30
- Constructive Solid Geometry, 152
- continuity
  - defined for 2D, 40
  - defined for 3D, 76
- continuous function, 76
- contour lines, 199
- contour rings, 9, 24, 137
- CoordAligner, 3, 174, 194, 202, 213, 224
  - defined, 173
  - figure, 233–235
- coordinate system, 173
  - handedness, 187
- coronal
  - defined, 212
- coronal orientation
  - defined, 212
- corrugated
  - appearance, 28
- corrugation
  - of surface, 118
- cortico-cortical projections, 329
- cranial direction, 187
- Cranial–Caudal axis, 214
- Cranio–Caudal Axis
  - defined, 187
- crease lines
  - of triangles, 89
- crest ridge, 210
- CSF, 145, 202
- CSG, 152
- CT, 205, 206
- CUBE Architecture
  - of Kaufman, 26
- cuberille, 25
- cuberille voxel, 26
- cuberille voxel faces, 26
- cuboid voxel, 77
- culling, 153
- current dipole moment, 179
- curvature extremum, 197
- curvature maxima, 196
- curvature maximum, 188
- Cyberware Laboratories, Inc., 197
- cycle
  - of vertices, 93
- DC, 114
- Delaunay triangles, 293
- Delaunay triangularization, 292
- dendritic tree, 135, 180
- Depth Complexity
  - for SpiderWeb surfaces, 115
- depth complexity
  - defined, 114
  - figure, 117
- depth-first, 153
- depth-first search, 81
- descending vertex values, 137
- digitization errors, 240
- Direct Method, 249
- disambiguate, 63
- disambiguates, 27
- disconnected system of hits, 163
- disjoint triangles, 30
- distinct, 130
- Dividing Cubes, 8, 28
- Dolphin Imaging, Inc., 197
- downhill, 63
- dried skull, 27
- dynamic programming, 25
- edge adjacent voxels



- figure, 80
- EEG, 209, 314–316, 329
- EEG electrodes, 209
- ellipsoid of minimum resolvable curvature, 131
- emerging triangle edge, 98
- empty voxel face, 157
- Euclidean 3-space, 153
- Euclidian metric
  - defined, 154
- Euler angles, 199
- Euler operators, 129, 152
- Euler primitives, 152
- Euler sums
  - defined, 130
- Eulers-Poincarè law
  - defined, 129
- evenfurcate, 89
- excessive triangles, 118
- extended voxel cases, 87
- extensive-source model, 318
- exterior path, 77, 78
- extremal points, 196
- extremum
  - magnetic field, 175
- face adjacent voxels
  - figure, 80
- face normal
  - of triangle, 93
  - figure, 95
- face-wise continuous, 150
- facial average value, 89
- fidelity, 22, 24, 27, 31, 33, 267, 335
- fiducial mark
  - defined, 188
- finite element analysis, 23, 29
- fission, 144
  - figure, 141
- fission face, 137
- fissions, 145
- flip angle, 184
- floating-point round-off error, 28
- foam, 162
- frequency encoding direction, 185
- full-resolution model, 31
- fuse, 137
- fuses, 145
- fusion, 144
  - figure, 141
- fusion face, 137
- fuzzy edges, 27
- fuzzy gradient, 32
- Gamma Knife, 173
- gaps
  - in surface tiles, 85
- Gaussian curvature, 56
- gaussian noise, 205
- genus, 116, 129
- genus change, 137
- glue, 129
- good segmentation, 137
- Gouraud's smoothing algorithm, 26
- gradient magnetic field, 184
- grain, 118
- Gram-Schmidt orthonormalization, 213
- GYROSCAN, 12, 187, 278
- half-space, 162
- handedness, 187
- hat, 198
- head based coordinate system, 187
- head-based coordinate system, 187
- head-frame coordinate system
  - figure, 239
- heads-up display, 206
- heart
  - biomagnetic fields, 175
- hedge

- our bet, 75
- hedgehog mode, 106
  - figure, 111
- hedgehog triangle normal mode, 94
- Herman80, 28
- hit
  - defined, 81
  - figure, 42
- hit vertices, 98
- hits, 9, 12, 35
- holes
  - between voxels, 85
  - in surface tiles, 85
- homeomorphic, 155, 156
- hot wire, 25
- Hough transform, 202, 203, 216
  - defined, 202
- hyperbolic curve, 40
- hyperbolic sheet, 76
- ill conditioned matrices, 242
- implicit functions, 29
- impossible surface configuration, 47
  - figure, 47
- inconsistent equations, 240
- incorrect topologies, 32
- infinite distance, 77
- inside
  - defined, 77
- internal body cavities, 31
- interpolant function, 40
- interpolated edges, 66
- interpolated vertices, 66
- interpolation artifacts, 55
- irregular face, 163
- irregular voxel face, 157
- ISG, Inc., 25, 26
- Jaszczak phantom, 201
- Jordan curve, 48, 52
- Jordan curve theorem
  - figure, 72
- Jordan separation theorem, 71
- Jordan surface, 24, 27, 33
- Jordan surface theorem, 90, 144
- Josephson junction, 177
- Joule–Thompson refrigerator, 177
- Kirchoff’s law, 180
- kiss, 48, 71
- Klein bottle, 155
- knitting, 63, 153
- knitting rules, 81
- Larmor resonance frequency, 184
- lattice
  - defined, 77
- law of Biot and Savart, 179
- least squares solution, 242
- Left Hand coordinate system
  - defined, 187
- left-handed coordinate system, 214
- Left-Right Axis, 214
- Left-Right axis
  - defined, 187
- legitimate surface, 30, 103
- legs
  - of triangle, 93
- line vertex, 40
- linear interpolation, 40
- liquid helium, 177
- longitudinal fissure, 216
  - defined, 202
- lookup tables, 87
- Möbius condition, 90
- Möbius rule, 91
- macroscopic, 150
- macroscopic properties
  - summarized, 150
- Magnetic Resonance Imaging, 3
- Magnetic Source Imaging, 3, 174

- manifold, 153, 156, 292
- manifold surface, 151
  - defined, 93
- Marching Cubes, 28, 34, 103
- marching cubes, 30, 87
- measurement surface, 292
- MEG, 173, 178, 202, 213, 292, 314–316
  - defined, 173
- microscopic properties
  - summarized, 150
- midsagittal plane, 203, 204, 212, 320
  - defined, 212
- Möbius strip, 155
- monkey brain slices, 29
- Moore Penrose pseudo-inverse, 242
- Morse theory, 25
- mouses, 73
- Movie BYU, 24
- MRI, 3, 138, 173, 174, 186, 195, 205, 214, 224–226, 228–231, 244–248, 283, 284, 286, 329
- MRI magnet bore, 187
- MSI, 3, 174, 175, 182, 185–189, 195, 204, 214, 304
  - defined, 173
- multi-resolution analysis, 137
- multi-resolution modeling, 31
- nasion, 13, 198, 209, 213, 217, 223, 228, 229, 234, 236, 239, 244, 249, 250, 253, 267
  - defined, 210
- Nasion Pointer Traces
  - figure, 236
- Nasion Sagittal Slice
  - figure, 229
- neural net, 25
- neuromagnetometer
  - figure, 176
- NMR, 182
- No more 4-hit faces rule, 55
- noise, 20, 32, 71, 127
  - alpha rhythm, 316
  - figure, 21
- noisy artifacts
  - figure, 138
- noisy images, 244
- normal resolution phase, 161
- normal resolution process, 161
- normal vector
  - of triangle, 93
- Nuclear Magnetic Resonance, 182
- numerical simulation experiments, 55
- Nyquist, 66
- Nyquist length, 66, 71, 131, 165
- object genus, 136
- Occam's Razor, 165
- octree, 116
- open ball, 154, 166
- oriented, 93, 153, 156
- orthodontics, 197
- osculate, 48, 71
- osculation
  - figure, 144
- outside
  - defined, 77
  - direction, 77
- outward
  - direction, 77
- packed, 75
  - tiling, 85
- pair-furcate, 136
- parabolic curve, 40
- parallel, 30
  - processing, 26
- PARCUM II, 25
- parieto-occipital sulcus, 319, 320, 328, 330
- partial volume, 27



- periauricular point, 188
- Periauricular Pointer Traces
  - figure, 237
- periauricular points, 209
  - defined, 210
- perturbation assumption, 157
- PET, 173, 199, 205, 228
- phase encoding direction, 185
- phase encoding techniques, 182
- Philips Gyroscan, 249, 256
- Philips Medical Systems, Inc., 12
- Phillips Gyroscan MRI system, 187
- Phoenix Data Systems, Inc., 24
- photogrammetry, 197
- pigeon-hole principle, 160
- pipped, 156
- Pixar, 27
- pixel
  - defined, 40, 77, 256
- pixel size, 256
- Pixel Slice
  - Table, 256
- pixel-point
  - voxel, 114
- pixel-spanning
  - voxel, 114
- PixelPicker
  - program, 224
- pixels, 12, 35
- pleats, 93
- plunging triangle edge, 98
- point singularity, 40
- pointer
  - configuration, 190
- Pointer Method, 249
- Pointer technique, 199, 202, 210, 216
- pointer technique, 198, 244, 266
- Pointers
  - being worn
  - figure, 193
  - use, 240
- Pointers Drawing
  - figure, 231
- Pointers Model
  - figure, 191, 192
- Pointers Side View
  - figure, 232
- Pointers technique, vii, 228
- poise, 199
- poles, 136
- Polhemus digitizer probe, 249
- Polhemus Navigation, Inc., 189, 197
- polygonization of implicit function surfaces, 28
- Powell's method, 213
- PPN
  - Computing, 215
- PPN head-based coordinate system, 188
- PPN Headframe, 188, 213, 214
- PPN transform, 213
- preauricular point, 188
- preauricular points, 209
- precession, 184
- precise, 252
- Procrustes transformation, 213
- proof by enumeration, 103
- Proton Emission Tomography, 199
- radius of curvature, 71
- re-entrant surface, 82
- recursive nature, 161
- Reeb graphs, 25
- region of interest, 114, 150
- regular face, 153, 161
- regular voxel face, 157
- rephasing, 184
- residual error, 213
- Residue theorem, 136
- Right Hand coordinate system, 187

- right hand coordinate system, 214
- RISC processor, 26
- saddle point, 210
  - figure, 210
- sagittal, 212
  - defined, 212
- scalp, 73
- scan conversion, 28
- scene, 114
- Schenker analysis, 137
- scraped clean, 153
- second-order gradiometer, 177
- seed  $\tau$  value, 137
- seed boxel, 36
- seed voxel face, 153
- segmentation, 145
- segments, 73
- SGI VGX, 30
- Shannon, 66
- sighting
  - defined, 188
- signed hits, 28
- simplex
  - defined, 93
- simplicial complex, 93, 152, 292
  - defined, 93, 155
- simply connected, 75
- single-dipole model, 318
- singular face, 136
- singular faces, 56
- singular value decomposition, 240
- singularities, 40, 76, 136, 144, 240
- singularity, 59, 127, 242, 243
- sinogram, 25
- skinny polygons, 29
- slice
  - defined, 77
- slice factor, 185, 256
- slice interval, 256
- slice selection direction, 185
- slice thickness, 185
- slop, 222
- Slop Distance Error
  - figure, 218
- slop error distance, 217
- smooth
  - defined, 155
- smoothness
  - defined for 2D, 39
  - defined for 3D, 76
- sonic digitizer, 197
- SPECT, 205, 228
- Spider In Web Method Method, 249
- spider web, 89
- SpiderWeb, 3, 173
- SpiderWeb A, 153
- SpiderWeb A algorithm, 153
- SpiderWeb algorithm, 150
- SpiderWeb B, 153
- SpiderWeb B algorithm, 153
- SpiderWeb isosurface system, 174
- SpiderWeb surface, 136
- spin contour lines, 7
- spin triangle mesh, 7
- spin triangles, 85
- spindles, viii
- spinning, 12
- spinning rule, 7
- spinning triangles, 7, 82, 150, 163
  - defined, 98
- SQUID, 177, 189, 292
  - defined, 174
- stereophotogrammetry, 196
- steriotactic head frame, 216
- straddle, 41, 157
- stride
  - defined, 78
- strong gradient, 69
- strong surface

- defined, 69
- sulcus, 29
- surface
  - defined, 154
  - surface construction, 114
  - surface rendering, 23
  - symmetry, 199
- T-mesh run
  - defined, 125
- ten-twenty system, 188, 209
- tessellation, 34, 75
- tessellation theory, 30
- tetrahedral decomposition, 30, 58
- tetrahedralization, 30
- thalamic connections, 329
- Thinking Machines, 30
- tiling errors, 8
  - figure, 10, 11
- topological polyhedra, 152
- topological polyhedron
  - defined, 93
- tragus, 209
- translation, 199
- triangle
  - defined, 93
- triangle articulating vertex, 98
- Triangle circulation
  - figure, 92
- triangle decimation, 31
- triangle face
  - defined, 93
- triangle face normal
  - defined, 93
- triangle mesh, 93
- triangle spinning, 75
- triangle spinning orientation, 153
- triangle spinning rule, 8, 9, 85, 87, 103
- triangle voxel face edge, 98
- triangulable
  - defined, 93
- trifurcate, 89, 136, 166
- trilinear interpolation, 30
- two manifold surface
  - defined, 155
- two sided, 156
  - surface, 155
- two way street, 91
- ultrasound, 206
- unbounded
  - defined, 155
- uncertainty, 153
- unresolved edge, 98
- uphill, 63
- V-block registration device, 206
- vertex, 35
- vertex adjacent voxels
  - figure, 80
- vertex normal
  - of triangle, 93
  - figure, 95
- vertex plucking, 118
- vertices
  - of voxel, 77
- virtual processors, 30
- Vitamin E, 198, 225, 228
- VO, 114
- VOI, 200
- volume gradient, 28
- volume occupancy
  - defined, 114
  - figure, 117
- Volume of Interest, 200
- volume rendering, 114
- volume visualization, 23
- Voronoi diagram, 293
- voxel, 36
  - defined, 77
  - definition



- figure, 78
- voxel circulation, 12
- voxel edge
  - defined, 77
  - figure, 78
- voxel face
  - figure, 78
- voxel face-wise continuity rule
  - defined, 87
  - figure, 88
- voxel faces
  - defined, 77
- voxel facewise continuity rule, 75
- voxel vertex
  - figure, 78
- voxel vertex state, 77
- voxels, 23
- waist triangles
  - defined, 103
- warpage error percentage, 219
- Warpage Percent Error
  - figure, 220
- weak
  - surface, 127
- Weaving Wall algorithm, 28
- webbing, 89
- winged edge, 31
- wobble, 184
- Wobble Distance Error
  - figure, 221
- wobble error, 219
- writhe
  - surface, 127
- wrong knitting, 74
- X axis, 214
- X-ray, 175
- X-ray refraction, 135
- z axis, 214
- zig-zag, 24
- zig-zag ordering
  - T-mesh, 125